

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 982 884**

51 Int. Cl.:

H04N 19/186 (2014.01)

H04N 19/86 (2014.01)

H04N 19/176 (2014.01)

H04N 19/117 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **24.08.2020 PCT/CN2020/110914**

87 Fecha y número de publicación internacional: **04.03.2021 WO21037004**

96 Fecha de presentación y número de la solicitud europea: **24.08.2020 E 20858263 (5)**

97 Fecha y número de publicación de la concesión europea: **10.04.2024 EP 3868101**

54 Título: **Un codificador, un decodificador y métodos correspondientes para realizar el desbloqueo de croma para bloques que usan la codificación de croma conjunta**

30 Prioridad:

23.08.2019 WO PCT/EP2019/072643

16.09.2019 WO PCT/RU2019/000639

07.10.2019 WO PCT/EP2019/077057

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
18.10.2024

73 Titular/es:

**HUAWEI TECHNOLOGIES CO., LTD. (100.0%)
Huawei Administration Building, Bantian,
Longgang District
Shenzhen, Guangdong 518129, CN**

72 Inventor/es:

**KOTRA, ANAND MEHER;
ALSHINA, ELENA ALEXANDROVNA;
ESENLIK, SEMIH;
WANG, BIAO;
GAO, HAN y
CHERNYAK, ROMAN IGOREVICH**

74 Agente/Representante:

SÁNCHEZ SILVA, Jesús Eladio

ES 2 982 884 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

DESCRIPCIÓN

Un codificador, un decodificador y métodos correspondientes para realizar el desbloqueo de croma para bloques que usan la codificación de croma conjunta

5

Referencia cruzada a solicitudes relacionadas

Esta solicitud de patente reivindica el beneficio de prioridad a la solicitud internacional n.º PCT/EP2019/072643, presentada el 23 de agosto de 2019, solicitud internacional n.º PCT/RU2019/000639, presentada el 16 de septiembre de 2019, y la solicitud internacional n.º PCT/EP2019/077057, presentada el 7 de octubre de 2019.

10

Campo técnico

Las realizaciones de la presente descripción generalmente se refieren al campo del procesamiento de imágenes, por ejemplo, a la codificación de imágenes fijas y/o imágenes de vídeo. Especialmente, la invención se refiere a las mejoras del filtro de desbloqueo.

15

Antecedentes

La codificación (codificación y decodificación) de imágenes se usa en una amplia gama de aplicaciones de imágenes digitales, por ejemplo, la transmisión de televisión digital, la transmisión de vídeo por internet y redes móviles, aplicaciones de conversaciones en tiempo real tales como los chats de vídeo y las videoconferencias, los DVD y los discos de Blu-ray, los sistemas de adquisición y edición de contenido de vídeo y las videocámaras de aplicaciones de seguridad. A partir del desarrollo del enfoque de codificación de vídeo híbrido basado en bloques en el estándar H.261 en 1990, se desarrollaron nuevas técnicas y herramientas de codificación de vídeo que formaron la base para los nuevos estándares de codificación de vídeo. Uno de los objetivos de la mayoría de los estándares de codificación de vídeo era lograr tasas de bits más bajas que las de sus predecesores sin sacrificar la calidad de la imagen. Otros estándares de codificación de vídeo comprenden vídeo MPEG-1, vídeo MPEG-2, ITU-T H.262/MPEG-2, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, codificación de vídeo avanzada (AVC), ITU-T H.265, codificación de vídeo de alta eficiencia (HEVC), ITU-T H.266/codificación de vídeo versátil (VVC) y extensiones, p. ej., escalabilidad y/o extensiones tridimensionales (3D), de estas normas. Los esquemas de codificación de imágenes basados en bloques tienen en común que, a lo largo de los bordes de los bloques, pueden aparecer artefactos de borde. Estos artefactos se deben a la codificación independiente de los bloques de codificación. Estos artefactos de borde frecuentemente son fácilmente visibles para el usuario. Un objetivo de la codificación de imágenes basada en bloques es reducir los artefactos de borde por debajo de un umbral de visibilidad. Esto se hace realizando un filtrado de bucle, por ejemplo, realizando un filtrado de desbloqueo. Dicho filtrado de desbloqueo se realiza, por un lado, en el lado de la decodificación para eliminar los artefactos de borde visibles, pero también en el lado de la codificación, para evitar que los artefactos de borde se codifiquen en la imagen. El proceso de filtro de desbloqueo generalmente incluye el proceso de decisión y el proceso de filtrado para los bordes de bloques de luma, y el proceso de decisión y el proceso de filtrado para bordes de bloques de croma.

20

25

30

35

40

Sin embargo, desbloquear un borde del bloque de croma entre dos bloques de croma adyacentes puede ser un desafío. Por ejemplo, la información utilizada para desbloquear el borde del bloque de croma puede derivarse del bloque de luma correspondiente y perderse durante el proceso de derivación, lo que conduce a un proceso de desbloqueo inexacto. En otro ejemplo, un nuevo tipo de bloques de croma, tal como los bloques de croma codificados con herramientas de codificación conjunta de Cb-Cr (JCCR) (o, a saber, herramientas de codificación conjunta de residuos de croma [JCCR, en inglés]) pueden plantear nuevos desafíos para el filtrado de desbloqueo.

45

Resumen

En vista de los desafíos mencionados anteriormente, la presente invención tiene como objetivo mejorar el filtrado de desbloqueo convencional. La presente invención tiene el objetivo de proporcionar un método de desbloqueo para desbloquear un borde del bloque de croma en un método de codificación o decodificación de imágenes, un aparato de filtro de desbloqueo para usar en un codificador o decodificador de imágenes y un producto de programa informático, que pueda realizar el filtrado de desbloqueo de forma precisa, por lo que el filtrado de desbloqueo debería ser más eficiente. Las realizaciones de la invención se definen según las características de las reivindicaciones independientes, y otras implementaciones ventajosas de las realizaciones se definen según las características de las reivindicaciones dependientes.

50

55

La base de la invención reivindicada puede encontrarse en la parte de la descripción relativa a la segunda realización de la presente invención. Otras realizaciones no se incluyen en la redacción de las reivindicaciones, pero se consideran útiles para entender la invención.

60

Breve descripción de los dibujos

En las siguientes realizaciones de la invención se describen con más detalle con referencia a las figuras y dibujos adjuntos, en los que:

65

la Figura 1A es un diagrama en bloque que muestra un ejemplo de un sistema de codificación de vídeo configurado para implementar realizaciones de la invención;

5 la Figura 1B es un diagrama en bloque que muestra otro ejemplo de un sistema de codificación de vídeo configurado para implementar realizaciones de la presente invención;

la Figura 2 es un diagrama en bloque que muestra un ejemplo de un codificador de vídeo configurado para implementar realizaciones de la invención;

10 la Figura 3 es un diagrama en bloque que muestra un ejemplo de estructura de un decodificador de vídeo configurado para implementar realizaciones de la invención;

la Figura 4 es un diagrama en bloque que ilustra un ejemplo de un aparato de codificación o un aparato de decodificación;

15 la Figura 5 es un diagrama en bloque que ilustra otro ejemplo de un aparato de codificación o un aparato de decodificación;

la Figura 6A muestra dos bloques de imágenes ilustrativos (tales como bloques de transformada o bloques de codificación);

20 la Figura 6B muestra dos bloques de imágenes ilustrativos (tales como bloques de transformada o bloques de codificación);

la Figura 7A es un diagrama conceptual que ilustra las ubicaciones relativas nominales verticales y horizontales de las muestras de luma y croma;

25 la Figura 7B es un diagrama esquemático que ilustra un bloque de luma y un bloque de croma coubicados;

la Figura 8 muestra dos bloques de luma ilustrativos (tales como componentes de luma de bloques de transformada o bloques de codificación);

30 las Figuras 9A a 9H son diagramas esquemáticos que ilustran un mecanismo ilustrativo desbloqueo en un borde del bloque (903, 913, 923, 933, 943, 953, 963, 973) de croma entre un primer bloque (901, 911, 921, 931, 941, 951, 961, 971) de croma de un primer bloque (601, 601') de imagen y un segundo bloque (902, 912, 922, 932, 942, 952, 962, 972) de croma de un segundo bloque (602, 602') de imagen;

35 la Figura 10 muestra una realización del dispositivo de filtro de desbloqueo según las realizaciones de la invención;

la Figura 11 muestra una realización de un diagrama de flujo que ilustra un método de desbloqueo para desbloquear un borde (903, 913, 923, 933, 943, 953, 963, 973) del bloque de croma entre un primer bloque (901, 911, 921, 931, 941, 951, 961, 971) de croma de un primer bloque (601, 601') de imagen y un segundo bloque (902, 912, 922, 932, 942, 952, 962, 972) de croma de un segundo bloque (602, 602') de imagen;

40 las Figuras 12A y 12B muestran dos tablas de mapeo de QP de croma ilustrativas;

la Figura 13 muestra una tabla de mapeo de Qp de croma separada ilustrativa para cada componente;

45 la Figura 14 es un diagrama en bloque que muestra una estructura ilustrativa de un sistema 3100 de suministro de contenidos que realiza un servicio de entrega de contenidos;

la Figura 15 es un diagrama en bloque que muestra una estructura de un dispositivo terminal ilustrativo.

50 la Figura 16 es un diagrama de flujo de un método de desbloqueo según algunos aspectos de la presente descripción; y

la Figura 17 es un diagrama de flujo de un proceso de decisión según algunos aspectos de la presente invención;

55 En los siguientes signos de referencia idénticos se refieren características idénticas o al menos funcionalmente equivalentes si no se especifican explícitamente lo contrario.

Descripción detallada de las realizaciones

60 La siguiente definición es para la referencia:

bloque de codificación: un bloque MxN de muestras para algunos valores de M y N, de modo que la división de un CTB en bloques de codificación es una partición.

65 bloque de árbol de codificación (CTB): Un bloque de NxN de muestras para algunos valores de N, de modo que la división de un componente en CTB es una partición.

unidad de árbol de codificación (CTU): Un CTB de muestras de luma, dos CTB correspondientes de muestras de croma de una imagen que tiene tres matrices de muestras o un CTB de muestras de una imagen monocromática o de una imagen que se codifica usando tres planos de color separados y estructuras de sintaxis utilizadas para codificar las muestras.

5 unidad de codificación (CU): Un bloque de codificación de muestras de luma, dos correspondientes bloques de codificación de muestras de croma de una imagen que tiene tres matrices de muestras o un bloque de codificación de muestras de una imagen monocromática o de una imagen que se codifica usando tres planos de color separados y estructuras de sintaxis utilizadas para codificar las muestras.

10 componente: Una matriz o una sola muestra de una de las tres matrices (luma y dos cromas) que componen una imagen en formato de color 4:2:0, 4:2:2 o 4:4:4 o la matriz o una sola muestra de la matriz que compone una imagen en formato monocromático.

15 En la siguiente descripción, se hace referencia a las figuras adjuntas, que forman parte de la descripción, y que muestran, a manera de ilustración, aspectos específicos de realizaciones de la invención o aspectos específicos en los que pueden usarse realizaciones de la presente invención. Se entiende que las realizaciones de la invención pueden usarse en otros aspectos y comprender cambios estructurales o lógicos no representados en las figuras. La siguiente descripción detallada, por lo tanto, no debe tomarse en un sentido limitativo, y el alcance de la presente invención está definida por las reivindicaciones adjuntas.

20 Por ejemplo, se entiende que una descripción en relación con un método descrito también puede ser válida para un dispositivo o sistema correspondiente configurado para realizar el método y viceversa. Por ejemplo, si se describen una o una pluralidad de etapas del método específico, un dispositivo correspondiente puede incluir una o una pluralidad de unidades, por ejemplo, unidades funcionales, para realizar la una o varias etapas del método descritas (por ejemplo, una unidad que realiza la una o varias etapas, o una pluralidad de unidades cada una que realiza una o más de la pluralidad de etapas), incluso si dicha una o más unidades no se describen o ilustran explícitamente en las figuras. Por otro lado, por ejemplo, si se describe un aparato específico basado en una o una pluralidad de unidades, por ejemplo, unidades funcionales, un método correspondiente puede incluir una etapa para realizar la funcionalidad de una o varias unidades (p. ej., una etapa que realiza la funcionalidad de una o varias unidades, o una pluralidad de etapas, cada una de las cuales realiza la funcionalidad de una o más de la pluralidad de unidades), incluso si una o varias etapas no se describen o ilustran explícitamente en las figuras. Además, se entiende que las características de las diversas realizaciones y/o aspectos ilustrativos descritos en la presente memoria pueden combinarse entre sí, a menos que se indique específicamente lo contrario.

35 La codificación de vídeo se refiere normalmente al procesamiento de una secuencia de imágenes que forman un vídeo o una secuencia de vídeo. En el campo de la codificación de vídeo, el término “cuadro” puede usarse como sinónimo de los términos “trama” o “imagen”. La codificación de vídeo (o la codificación en general) comprende dos partes: una codificación de vídeo y una decodificación de vídeo. La codificación de vídeo se realiza en el lado de origen, que normalmente comprende procesar (p. ej., comprimir) las imágenes de vídeo originales para reducir la cantidad de datos requeridos para representarlas (para un almacenamiento y/o una transmisión más eficiente[s]). La decodificación de vídeo se realiza en un lado de destino y normalmente comprende el procesamiento inverso en comparación con el codificador para reconstruir las imágenes de vídeo. Se entenderá que las realizaciones que se refieren a la “codificación” de imágenes de vídeo (o imágenes en general) se refieren a la “codificación” o la “decodificación” de imágenes de vídeo o de unas respectivas secuencias de vídeo. La combinación de la parte de codificación y la parte de decodificación también se denomina CÓDEC (combinación de los términos codificación y decodificación). En el caso de una codificación de vídeo sin pérdidas, las imágenes de vídeo originales pueden reconstruirse, es decir, las imágenes de vídeo reconstruidas tienen la misma calidad que las imágenes de vídeo originales (suponiendo que no haya ninguna pérdida de transmisión ni ninguna otra pérdida de datos durante el almacenamiento o la transmisión). En el caso de una codificación de vídeo con pérdidas, se realiza una compresión adicional, p. ej., por cuantificación, para reducir la cantidad de datos que representan las imágenes de vídeo, que no pueden reconstruirse completamente en el decodificador, es decir, la calidad de las imágenes de vídeo reconstruidas es menor o peor en comparación con la calidad de las imágenes de vídeo originales.

55 Varios estándares de codificación de vídeo pertenecen al grupo de “códecs de vídeo híbridos con pérdidas” (es decir, combinan una predicción espacial y una predicción temporal en el dominio de muestra y una codificación por transformada 2D para aplicar una cuantificación en el dominio de transformada). Cada imagen de una secuencia de vídeo se divide típicamente en un conjunto de bloques no superpuestos y la codificación se realiza típicamente en un nivel de bloque. En otras palabras, el vídeo normalmente se procesa en el codificador, es decir, se codifica, a nivel de bloque (bloque de vídeo), p. ej., usando una predicción espacial (intraimagen) y/o una predicción temporal (interimagen) para generar un bloque de predicción, restando el bloque de predicción del bloque actual (el bloque que se está procesando actualmente o que hay que procesar) para obtener un bloque residual, transformando el bloque residual y cuantificando el bloque residual en el dominio de transformada para reducir la cantidad de datos a transmitir (compresión), mientras que, en el decodificador, al bloque codificado o comprimido se le aplica el procesamiento inverso en comparación con el del codificador para reconstruir el bloque actual para su representación. Además, el codificador duplica el bucle de procesamiento del decodificador de modo que ambos generarán predicciones idénticas (p. ej., intrapredicciones e interpredicciones) y/o reconstrucciones para procesamiento, es decir, codificación, los bloques posteriores.

En las siguientes realizaciones de un sistema 10 de codificación de vídeo se describen un codificador 20 de vídeo y un decodificador 30 de vídeo basándose en las Figuras 1 a 3.

5 La Figura 1A es un diagrama de bloques esquemático que ilustra un sistema 10 de codificación de ejemplo, p. ej., un sistema 10 de codificación de vídeo (o sistema 10 de codificación corta), que puede utilizar técnicas de esta presente solicitud. El codificador 20 de vídeo (o codificador corto 20) y el decodificador 30 de vídeo (o decodificador corto 30) del sistema 10 de codificación de vídeo son ejemplos de dispositivos que pueden configurarse para realizar técnicas según diversos ejemplos descritos en la presente solicitud.

10 Tal y como se muestra en la Figura 1A, el sistema 10 de codificación comprende un dispositivo 12 de origen configurado para proporcionar unos datos 21 de imagen codificados a, por ejemplo, un dispositivo 14 de destino para decodificar los datos 13 de imagen codificados.

15 El dispositivo 12 de origen comprende un codificador 20 y, adicionalmente, es decir, opcionalmente, puede comprender una fuente 16 de imágenes, un preprocesador (o unidad de preprocesamiento) 18, p. ej., un preprocesador 18 de imágenes, y una interfaz 22 de comunicación o unidad 22 de comunicación.

20 La fuente 16 de imágenes puede comprender o ser cualquier tipo de dispositivo de captura de imágenes, por ejemplo, una cámara, para capturar una imagen del mundo real y/o, cualquier tipo de dispositivo de generación de imágenes, por ejemplo, un procesador de gráficos de ordenador, para generar una imagen animada por ordenador o cualquier tipo de otro dispositivo para obtener y/o proporcionar una imagen del mundo real, una imagen generada por ordenador (p. ej., un contenido de pantalla o una imagen de realidad virtual [RV]) y/o cualquier combinación de las mismas (p. ej., una imagen de realidad aumentada [RA]). La fuente de imágenes puede ser cualquier tipo de memoria o de almacenamiento que almacene cualquiera de las imágenes mencionadas anteriormente.

25 A diferencia del preprocesador 18 y del procesamiento realizado por la unidad 18 de preprocesamiento, la imagen o los datos 17 de imagen también pueden denominarse imagen sin procesar o datos 17 de imagen sin procesar. El preprocesador 18 está configurado para recibir los datos 17 de imagen (sin procesar) y para realizar su preprocesamiento en los datos 17 de imagen a fin de obtener una imagen preprocesada 19 o unos datos 19 de imagen preprocesados. El preprocesamiento realizado por el preprocesador 18 puede comprender, p. ej., un recorte, una conversión de formato de color (p. ej., de RGB a YCbCr), una corrección de color o una eliminación del ruido. Puede entenderse que la unidad 18 de preprocesamiento puede ser un componente opcional.

30 El codificador 20 de vídeo está configurado para recibir los datos 19 de imagen preprocesados y proporcionar unos datos 21 de imagen codificados (más adelante se darán más detalles basándose en, por ejemplo, la Figura 2).

35 La interfaz 22 de comunicación del dispositivo 12 de origen puede configurarse para recibir los datos 21 de imagen codificados y transmitir los datos 21 de imagen codificados (o cualquier otra versión procesada de los mismos) por un canal 13 de comunicación a otro dispositivo, p. ej., el dispositivo 14 de destino o cualquier otro dispositivo, para su almacenamiento o reconstrucción directa.

40 El dispositivo 14 de destino comprende un decodificador 30 (p. ej., un decodificador 30 de vídeo) y, adicionalmente, es decir, opcionalmente, puede comprender una interfaz 28 de comunicación o unidad 28 de comunicación, un posprocesador 32 (o unidad 32 de posprocesamiento) y un dispositivo 34 de visualización.

45 La interfaz 28 de comunicación del dispositivo 14 de destino está configurada para recibir los datos 21 de imagen codificados (o cualquier otra versión procesada de los mismos), p. ej., directamente del dispositivo 12 de origen o de cualquier otra fuente, p. ej., un dispositivo de almacenamiento, p. ej., un dispositivo de almacenamiento de datos de imagen codificados, y proporcionar los datos 21 de imagen codificados al decodificador 30.

50 La interfaz 22 de comunicación y la interfaz 28 de comunicación pueden configurarse para transmitir o recibir los datos 21 de imagen codificados o datos codificados 13 a través de un enlace de comunicación directa entre el dispositivo 12 de origen y el dispositivo 14 de destino, por ejemplo, una conexión directa cableada o inalámbrica, o a través de cualquier tipo de red, por ejemplo, una red cableada o inalámbrica o cualquier combinación de las mismas, o cualquier tipo de red privada y pública, o cualquier tipo de combinación de las mismas.

55 La interfaz 22 de comunicación puede estar, p. ej. configurada para empaquetar los datos 21 de imagen codificados en un formato apropiado, p. ej., unos paquetes, y/o para procesar los datos de imagen codificados usando cualquier tipo de codificación o procesamiento de transmisión para su transmisión por un enlace de comunicación o a través de una red de comunicación.

60 La interfaz 28 de comunicación, que forma la equivalente de la interfaz 22 de comunicación, puede estar, p. ej., configurada para recibir los datos transmitidos y procesar los datos de transmisión usando cualquier tipo de decodificación o procesamiento de transmisión y/o de desempaquetado correspondiente para obtener los datos 21 de imagen codificados.

65

Tanto la interfaz 22 de comunicación como la interfaz 28 de comunicación pueden configurarse como interfaces de comunicación unidireccionales, tal y como se indica mediante la flecha el canal 13 de comunicación de la Figura 1A, que apunta del dispositivo 12 de origen al dispositivo 14 de destino, o como interfaces de comunicación bidireccionales, y pueden configurarse para, p. ej., enviar y recibir mensajes, para, p. ej., establecer una conexión, para acusar recibo e intercambiar cualquier otra información relacionada con el enlace de comunicación y/o con la transmisión de datos, p. ej., una transmisión de datos de imagen codificados.

El decodificador 30 está configurado para recibir los datos 21 de imagen codificados y proporcionar unos datos 31 de imagen decodificados o una imagen 31 decodificada (más adelante se darán más detalles basándose en, p. ej., la Figura 3 o la Figura 5).

El posprocesador 32 del dispositivo 14 de destino se configura para posprocesar los datos 31 de imagen descodificados (también denominados datos de imagen reconstruidos), por ejemplo, la imagen descodificada 31, para obtener datos 33 de imagen posprocesados, por ejemplo, una imagen posprocesada 33. El procesamiento posterior realizado por la unidad 32 de posprocesamiento puede comprender, por ejemplo, conversión de formato de color (por ejemplo, de YCbCr a RGB), corrección de color, recorte o remuestreo, o cualquier otro procesamiento, por ejemplo, para preparar los datos 31 de imagen descodificados para su visualización, por ejemplo, mediante el dispositivo 34 de visualización.

El dispositivo 34 de visualización del dispositivo 14 de destino se configura para recibir los datos 33 de imagen procesados posteriormente para visualizar la imagen, por ejemplo, a un usuario u observador. El dispositivo 34 de visualización puede ser o comprender cualquier tipo de pantalla para representar la imagen reconstruida, por ejemplo, una pantalla o monitor integrado o externo. Las pantallas pueden comprender, por ejemplo, pantallas de cristal líquido (LCD), pantallas de diodos orgánicos emisores de luz (OLED), pantallas de plasma, proyectores, pantallas de micro LED, cristal líquido sobre silicio (LCoS), procesador de luz digital (DLP) o cualquier tipo de pantalla.

Aunque en la Figura 1A se han representado el dispositivo 12 de origen y el dispositivo 14 de destino como dispositivos separados, algunas realizaciones de dispositivos también pueden comprender ambos o ambas funcionalidades, el dispositivo 12 de origen o una funcionalidad correspondiente, y el dispositivo 14 de destino o una funcionalidad correspondiente. En tales realizaciones, el dispositivo 12 de origen o la funcionalidad correspondiente y el dispositivo 14 de destino o la funcionalidad correspondiente pueden implementarse usando el mismo hardware y/o software o mediante hardware y/o software separado o cualquier combinación de los mismos.

Tal y como será evidente para el experto en la técnica basándose en la descripción, la existencia y la división (exacta) de funcionalidades de las distintas unidades o funcionalidades que hay dentro del dispositivo 12 de origen y/o del dispositivo 14 de destino tal y como se muestran en la Figura 1A pueden variar dependiendo del dispositivo y de la aplicación reales.

El codificador 20 (p. ej., un codificador 20 de vídeo) o el decodificador 30 (p. ej., un decodificador 30 de vídeo) o tanto el codificador 20 como el decodificador 30 pueden implementarse a través de un sistema de circuitos de procesamiento, tal y como se muestra en la Figura 1B, como uno o más microprocesadores, procesadores de señales digitales (DSP, en inglés), circuitos integrados de aplicación específica (ASIC, en inglés), matrices de puertas programables en campo (FPGA, en inglés), lógica discreta, hardware, codificación de vídeo dedicada o cualquier combinación de los mismos. El codificador 20 puede implementarse a través de una circuitería 46 de procesamiento para encarnar los diversos módulos que se mencionan con respecto al codificador 20 de la Figura 2 y/o cualquier otro sistema o subsistema codificador descrito en la presente memoria. El decodificador 30 puede implementarse a través de la circuitería 46 de procesamiento para encarnar los diversos módulos que se mencionan con respecto al decodificador 30 de la Figura 3 y/o cualquier otro sistema o subsistema decodificador descrito en la presente memoria. La circuitería de procesamiento puede configurarse para realizar las diversas operaciones que se mencionan más adelante. Tal y como se muestra en la Figura 5, si las técnicas se implementan parcialmente en software, un dispositivo puede almacenar unas instrucciones para el software en un medio de almacenamiento legible por ordenador no transitorio adecuado y puede ejecutar las instrucciones en hardware usando uno o más procesadores para realizar las técnicas de esta descripción. Cualquiera del codificador 20 de vídeo y el decodificador 30 de vídeo puede integrarse como parte de un codificador/decodificador (CÓDEC) combinado en un solo dispositivo, por ejemplo, tal y como se muestra en la Figura 1B.

El dispositivo 12 de origen y el dispositivo 14 de destino pueden comprender cualquiera de una amplia gama de dispositivos, incluido cualquier tipo de dispositivo portátil o estacionario, p. ej., un ordenador portátil o notebook, un teléfono móvil, un teléfono inteligente, una tableta u ordenador de tableta, una cámara, un ordenador de sobremesa, un decodificador, un televisor, un dispositivo de visualización, un reproductor de medios digitales, una consola de videojuegos, un dispositivo de transmisión de vídeo (tal como un servidor de servicios de contenido o un servidor de entrega de contenido), un dispositivo receptor de difusiones, un dispositivo transmisor de difusiones o dispositivos similares y puede usar ningún o cualquier tipo de sistema operativo. En algunos casos, el dispositivo 12 de origen y el dispositivo de destino 14 pueden estar equipados para la comunicación inalámbrica. Por lo tanto, el dispositivo 12 de origen y el dispositivo 14 de destino pueden ser dispositivos de comunicación inalámbrica.

En algunos casos, el sistema 10 de codificación de vídeo ilustrado en la Figura 1A es simplemente un ejemplo, y las técnicas de la presente solicitud pueden aplicarse a configuraciones de codificación de vídeo (p. ej., codificación de vídeo o decodificación de vídeo) que no incluyen necesariamente ninguna comunicación de datos entre los dispositivos de

codificación y de decodificación. En otro ejemplos, los datos se recuperan de una memoria local, se retransmiten en continuo por una red, o cosas similares. Un dispositivo de codificación de vídeo puede codificar y almacenar datos en una memoria, y/o un dispositivo de decodificación de vídeo puede recuperar y decodificar datos de una memoria. En algunos ejemplos, la codificación y la decodificación son realizadas por dispositivos que no se comunican entre sí, sino que simplemente codifican datos en una memoria y/o recuperan y decodifican datos de una memoria. Para facilitar la descripción, en la presente memoria se describen realizaciones de la invención, por ejemplo, por referencia a la codificación de vídeo de alta eficiencia (HEVC) o al software de referencia de codificación de vídeo versátil (VVC), el estándar de codificación de vídeo de próxima generación desarrollado por el Equipo de Colaboración Conjunta de Codificación de Vídeo (JCT-VC) del Grupo de Expertos en Codificación de Vídeo del ITU-T (VCEG) y el Grupo de Expertos en Imágenes de Movimiento (MPEG) de la ISO/IEC. Un experto en la técnica entenderá que las realizaciones de la invención no se limitan a HEVC o VVC.

Codificador y método de codificación

La Figura 2 muestra un diagrama de bloques esquemático de un codificador 20 de vídeo de ejemplo que está configurado para implementar las técnicas de la presente solicitud. En el ejemplo de la Figura 2, el codificador 20 de vídeo comprende una entrada 201 (o interfaz 201 de entrada), una unidad 204 de cálculo residual, una unidad 206 de procesamiento de transformada, una unidad 208 de cuantificación, una unidad 210 de cuantificación inversa, una unidad 212 de procesamiento de transformada inversa, una unidad 214 de reconstrucción, una unidad 220 de filtro de bucle, una memoria 230 intermedia de imágenes decodificadas (DPB, por sus siglas en inglés), una unidad 260 de selección de modo, una unidad 270 de codificación entrópica y una salida 272 (o interfaz 272 de salida). La unidad 260 de selección de modo puede incluir una unidad 244 de interpredicción, una unidad 254 de intrapredicción y una unidad 262 de partición. La unidad 244 de interpredicción puede incluir una unidad de estimación de movimiento y una unidad de compensación de movimiento (no mostrada). Un codificador 20 de vídeo como se muestra en la Figura 2 también puede denominarse codificador de vídeo híbrido o un codificador de vídeo según un códec de vídeo híbrido.

Puede decirse que la unidad 204 de cálculo residual, la unidad 206 de procesamiento de transformada, la unidad 208 de cuantificación y la unidad 260 de selección de modo forman un camino de señal directo del codificador 20, mientras que puede decirse que la unidad 210 de cuantificación inversa, la unidad 212 de procesamiento de transformada inversa, la unidad 214 de reconstrucción, la memoria intermedia 216, el filtro 220 de bucle, la memoria 230 intermedia de imágenes decodificadas (DPB), la unidad 244 de interpredicción y la unidad 254 de intrapredicción forman un camino de señal inverso del codificador 20 de vídeo, en donde el camino de señal inverso del codificador 20 de vídeo corresponde al camino de señal del decodificador (véase el decodificador 30 de vídeo de la Figura 3). También se dice que la unidad 210 de cuantificación inversa, la unidad 212 de procesamiento de transformada inversa, la unidad 214 de reconstrucción, el filtro 220 de bucle, la memoria 230 intermedia de imágenes decodificadas (DPB), la unidad 244 de interpredicción y la unidad 254 de intrapredicción forman el “decodificador incorporado” del codificador 20 de vídeo.

Imágenes y partición de imágenes (imágenes y bloques)

El codificador 20 puede configurarse para recibir, p. ej., por la entrada 201, una imagen 17 (o unos datos 17 de imagen), p. ej., una imagen de una secuencia de imágenes que forma un vídeo o una secuencia de vídeo. La imagen o los datos de imagen recibida(os) también puede(n) ser una imagen preprocesada 19 (o unos datos 19 de imagen preprocesados). En aras de simplicidad, la siguiente descripción se refiere a la imagen 17. La imagen 17 también puede denominarse imagen actual o imagen a codificar (en particular, en la codificación de vídeo, para distinguir la imagen actual de otras imágenes, p. ej., unas imágenes previamente codificadas y/o decodificadas de la misma secuencia de vídeo, es decir, la secuencia de vídeo que también comprende la imagen actual).

Una imagen (digital) es o puede considerarse como una matriz bidimensional o matriz de muestras con valores de intensidad. Una muestra en la matriz también puede denominarse píxel (forma corta del elemento de la imagen). El número de muestras en dirección horizontal y vertical (o eje) de la matriz o imagen define el tamaño y/o resolución de la imagen. Para la representación de color, normalmente se emplean tres componentes de color, es decir, la imagen puede representarse o incluir tres matrices de muestras. En el formato de RBG o el espacio de color, una imagen comprende una matriz de muestras roja, verde y azul correspondiente. Sin embargo, en la codificación de vídeo, cada píxel se representa normalmente en un formato o espacio de color de luminancia y crominancia, p. ej., YCbCr, que comprende una componente de luminancia indicada por Y (a veces también se usa L en su lugar) y dos componentes de crominancia indicadas por Cb y Cr. El componente Y de luminancia (o luma corta) representa el brillo o la intensidad del nivel de grises (por ejemplo, como en una imagen a escala de grises), mientras que los dos componentes Cb y Cr de crominancia (o croma corta) representan la cromaticidad o los componentes de información de color. En consecuencia, una imagen en formato YCbCr comprende una matriz de muestras de luminancia de valores de muestra de luminancia (Y), y dos matrices de muestras de crominancia de valores de crominancia (Cb y Cr). Las imágenes en formato RGB pueden convertirse o transformarse en formato YCbCr y viceversa, el proceso también se conoce como transformación de color o conversión. Si una imagen es monocromática, la imagen puede comprender solo una matriz de muestras de luminancia. Por consiguiente, una imagen puede ser, por ejemplo, una matriz de muestras de luma en formato monocromo o una matriz de muestras de luma y dos matrices correspondientes de muestras de croma en formato de color 4:2:0, 4:2:2 y 4:4:4.

Algunas realizaciones del codificador 20 de vídeo pueden comprender una unidad de partición de imagen (no representada en la Figura 2) configurada para partir la imagen 17 en una pluralidad de bloques 203 de imagen (que normalmente no se

solapan). Estos bloques también pueden denominarse bloques raíz, macrobloques (H.264/AVC), bloques de árbol de codificación (CTB, en inglés) o unidades de árbol de codificación (CTU, en inglés) (H.265/HEVC y VVC). La unidad de partición de imagen puede configurarse para usar el mismo tamaño de bloque para todas las imágenes de una secuencia de vídeo y la correspondiente cuadrícula que define el tamaño de bloque, o para cambiar el tamaño de bloque entre imágenes o subconjuntos o grupos de imágenes, y partir cada imagen en los bloques correspondientes.

En otras realizaciones, el codificador de vídeo puede configurarse para recibir directamente un bloque 203 de la imagen 17, p. ej., uno, varios o todos los bloques que forman la imagen 17. El bloque 203 de imagen también puede denominarse bloque de imagen actual o bloque de imagen a codificar.

Al igual que la imagen 17, el bloque 203 de imagen es nuevamente o puede considerarse otra vez como una matriz bidimensional o una matriz de muestras con valores de intensidad (valores de muestra), aunque de dimensiones más pequeñas que la imagen 17. En otras palabras, el bloque 203 puede comprender, p. ej., una matriz de muestras (p. ej., una matriz de luma en caso de una imagen monocromática 17 o una matriz de luma o croma en caso de una imagen de color) o tres matrices de muestras (p. ej., una matriz de luma y dos matrices de croma en caso de una imagen 17 de color) o cualquier otro número y/o tipo de matrices dependiendo del formato de color aplicado. Una cantidad de muestras en unas direcciones (o ejes) horizontal y vertical del bloque 203 define el tamaño del bloque 203. Por consiguiente, un bloque puede ser, por ejemplo, una matriz de $M \times N$ (M columnas por N filas) de muestras o una matriz de $M \times N$ de coeficientes de transformada.

Algunas realizaciones del codificador 20 de vídeo que se muestra en la Figura 2 pueden configurarse para codificar la imagen 17 bloque a bloque, p. ej., la codificación y la predicción se realizan por cada bloque 203.

Las realizaciones del codificador 20 de vídeo, tal como se muestra en la Figura 2, pueden configurarse además para dividir y/o codificar la imagen mediante el uso de segmentos (también denominados segmentos de vídeo), en donde una imagen puede dividirse o codificarse utilizando uno o más segmentos (típicamente sin superponerse), y cada segmento puede comprender uno o más bloques (p. ej., CTU) o uno o más grupos de bloques (p. ej., mosaicos (H.265/HEVC y VVC) o ladrillos (VVC)).

Las realizaciones del codificador 20 de vídeo, tal como se muestra en la Figura 2, pueden configurarse además para dividir y/o codificar la imagen mediante el uso de grupos de segmentos/mosaicos (también denominados grupos de mosaicos de vídeo) y/o mosaicos (también denominados mosaicos de vídeo), en donde una imagen puede dividirse o codificarse usando uno o más grupos de segmentos/mosaicos (típicamente sin superposición), y cada grupo de segmentos/mosaicos puede comprender, p. ej., uno o más bloques (p. ej., CTU) o uno o más mosaicos, en donde cada mosaico, p. ej., puede tener forma rectangular y puede comprender uno o más bloques (p. ej., CTU), p. ej., bloques completos o fraccionados.

Cálculo residual

La unidad 204 de cálculo residual puede configurarse para calcular un bloque residual 205 (denominado también residuo 205) basándose en el bloque 203 de imagen y en un bloque 265 de predicción (más adelante se dan más detalles acerca del bloque 265 de predicción), p. ej., restando unos valores de muestra del bloque 265 de predicción de unos valores de muestra del bloque 203 de imagen, muestra a muestra (píxel a píxel), para obtener el bloque residual 205 en el dominio de muestra.

Transformar

La unidad 206 de procesamiento de transformada puede configurarse para aplicar una transformada, p. ej., una transformada de coseno discreta (DCT) o una transformada de seno discreta (DST), a los valores de muestra del bloque residual 205 para obtener unos coeficientes 207 de transformada en un dominio de transformada. Los coeficientes 207 de transformada también se pueden denominar coeficientes residuales de transformación y representar el bloque residual 205 en el dominio de transformación.

La unidad 206 de procesamiento de transformada puede configurarse para aplicar unas aproximaciones enteras de la transformada DCT/DST, tales como las transformadas especificadas en el caso de H.265/HEVC. En comparación con una transformada DCT ortogonal, tales aproximaciones enteras normalmente se ajustan a escala por un factor específico. Para conservar la norma del bloque residual que se procesa usando transformadas directas e inversas, se aplican unos factores de ajuste de escala adicionales como parte del proceso de transformada. Los factores de ajuste de escala se eligen normalmente basándose en ciertas restricciones, por ejemplo, que los factores de ajuste de escala sean una potencia de dos para operaciones de desplazamiento, una profundidad de bits de los coeficientes de transformada, un compromiso entre la precisión y los costes de implementación, etc. En el caso de la transformada inversa, unos factores de ajuste de escala específicos son especificados por, p. ej., una unidad 212 de procesamiento de transformada inversa (y, en el caso de la transformada inversa correspondiente, por, p. ej., una unidad 312 de procesamiento de transformada inversa en un decodificador 30 de vídeo), y en el caso de la transformada directa, unos factores de ajuste de escala correspondientes pueden ser especificados por, p. ej., una unidad 206 de procesamiento de transformada en un codificador 20.

Algunas realizaciones del codificador 20 de vídeo (o, respectivamente, de la unidad 206 de procesamiento de transformada) pueden configurarse para dar salida a unos parámetros de transformada, p. ej., un tipo de transformada o de transformadas, p. ej., directamente o codificados o comprimidos a través de la unidad 270 de codificación entrópica, para que, p. ej., el decodificador 30 de vídeo los pueda recibir y usar para la decodificación.

5

Cuantificación

La unidad 208 de cuantificación puede configurarse para cuantificar los coeficientes 207 de transformada a fin de obtener unos coeficientes cuantificados 209, p. ej., aplicando una cuantificación escalar o una cuantificación vectorial. Los coeficientes cuantificados 209 también se pueden denominar coeficientes 209 de transformada cuantificados o coeficientes residuales cuantificados 209.

10

El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos o a todos los coeficientes 207 de transformada. Por ejemplo, durante la cuantificación, un coeficiente de transformada de n bits puede redondearse a la baja hasta dar un coeficiente de transformada de m bits, donde n es mayor que m . El grado de cuantificación puede modificarse ajustando un parámetro de cuantificación (QP, por sus siglas en inglés). Por ejemplo, en el caso de la cuantificación escalar, se puede aplicar un ajuste de escala diferente para lograr una cuantificación más fina o más basta. Tamaños de etapa de cuantificación más pequeños corresponden a una cuantificación más fina, mientras que tamaños de etapa de cuantificación más grandes corresponden a cuantificación más basta. El tamaño de etapa de cuantificación apropiado puede indicarse mediante un parámetro de cuantificación (QP). El parámetro de cuantificación puede ser, por ejemplo, un índice de un conjunto predefinido de tamaños de etapa de cuantificación aplicables. Por ejemplo, unos parámetros de cuantificación pequeños pueden corresponder a una cuantificación fina (tamaños de etapa de cuantificación pequeños) y unos parámetro de cuantificación grandes pueden corresponder a una cuantificación basta (tamaños de etapa de cuantificación grandes) o viceversa. La cuantificación puede incluir una división por un tamaño de etapa de cuantificación y una descuantificación correspondiente y/o la descuantificación inversa, p. ej., por parte de la unidad 210 de cuantificación inversa, puede incluir una multiplicación por el tamaño de etapa de cuantificación. Algunas realizaciones según algunos estándares, p.ej., el correspondiente a la codificación HEVC, pueden configurarse para usar un parámetro de cuantificación para determinar el tamaño de etapa de cuantificación. Generalmente, el tamaño de etapa de cuantificación puede calcularse basándose en un parámetro de cuantificación usando una aproximación de punto fijo de una ecuación que incluye una división. Se pueden introducir unos factores de ajuste de escala adicionales para la cuantificación y la descuantificación a fin de restablecer la norma del bloque residual, que puede resultar modificada debido al ajuste de escala usado en la aproximación de punto fijo de la ecuación para obtener el tamaño de etapa de cuantificación y el parámetro de cuantificación. En una implementación de ejemplo, se pueden combinar el ajuste de escala de la transformada inversa y la descuantificación. Alternativamente, pueden usarse unas tablas de cuantificación personalizadas y señalarse de un codificador a un decodificador, p. ej., en un flujo de bits. La cuantificación es una operación con pérdidas en la que la pérdida aumenta a medida que aumentan los tamaños de etapa de cuantificación.

15

20

25

30

35

Algunas realizaciones del codificador 20 de vídeo (o, respectivamente, de la unidad 208 de cuantificación) pueden configurarse para dar salida a unos parámetros de cuantificación (QP), p. ej., directamente o codificados a través de la unidad 270 de codificación entrópica, para que, p. ej., el decodificador 30 de vídeo los pueda recibir y aplicar para la decodificación.

40

Cuantificación inversa

La unidad 210 de cuantificación inversa está configurada para aplicar la cuantificación inversa de la unidad 208 de cuantificación a los coeficientes cuantificados para obtener unos coeficientes descuantificados 211, p. ej., aplicando la inversa del plan de cuantificación aplicado por la unidad 208 de cuantificación basándose en o usando el mismo tamaño de etapa de cuantificación que la unidad 208 de cuantificación. Los coeficientes descuantificados 211 también pueden denominarse coeficientes residuales descuantificados 211 y corresponden –aunque normalmente no son idénticos a los coeficientes de transformada debido a la pérdida debida a la cuantificación– a los coeficientes 207 de transformada.

50

Transformada inversa

La unidad 212 de procesamiento de transformada inversa está configurada para aplicar la transformada inversa de la transformada aplicada por la unidad 206 de procesamiento de transformada, p. ej., una transformada de coseno discreta (DCT) inversa o una transformada de seno discreta (DST) inversa u otras transformadas inversas, para obtener un bloque residual 213 reconstruido (o unos correspondientes coeficientes descuantificados 213) en el dominio de muestra. El bloque residual 213 reconstruido también puede denominarse bloque 213 de transformada.

55

Reconstrucción

60

La unidad 214 de reconstrucción (p. ej., un sumador 214) está configurada para añadir el bloque 213 de transformada (es decir, el bloque residual 213 reconstruido) al bloque 265 de predicción para obtener un bloque reconstruido 215 en el dominio de muestra, p. ej., sumando –muestra a muestra– los valores de muestra del bloque residual 213 reconstruido y los valores de muestra del bloque 265 de predicción.

65

Filtrado

La unidad 220 de filtro de bucle (o “filtro 220 de bucle” corto) se configura para filtrar el bloque reconstruido 215 a fin de obtener un bloque filtrado 221 o, en general, para filtrar muestras reconstruidas para obtener valores de muestra filtrada. La unidad de filtro de bucle está configurada para, p. ej., suavizar unas transiciones entre píxeles o, de lo contrario, mejorar la calidad de vídeo. La unidad 220 de filtro de bucle puede comprender uno o más filtros de bucle, tales como un filtro de desbloqueo, un filtro de desplazamiento adaptativo de muestras (SAO, en inglés) o uno o más filtros diferentes, p. ej., un filtro de bucle adaptativo (ALF, en inglés), un filtro de supresión de ruido (NSF, en inglés) o cualquier combinación de los mismos. En un ejemplo, la unidad 220 de filtro de bucle puede comprender un filtro de desbloqueo, un filtro SAO y un filtro ALF. El orden del proceso de filtrado puede ser el filtro de desbloqueo, SAO y ALF. En otro ejemplo, se añade un proceso denominado mapeo de luma con escala de croma (LMCS, en inglés) (es decir, el remodelador en bucle adaptativo). Este proceso se realiza antes del filtro de desbloqueo. En otro ejemplo, el proceso de filtro de desbloqueo también se puede aplicar a los bordes de subbloques internos, p. ej., los bordes de subbloques afines, los bordes de subbloques de ATMVP, los bordes de transformación de subbloques (SBT) y los bordes de intra subdivisión (ISP).

Para eliminar eficazmente los artefactos de bloque que se producen en “bloques” grandes, la codificación de vídeo versátil (VVC, en inglés) utiliza un filtro de desbloqueo de pulsaciones más prolongadas. En este caso, el término “bloques” se usa de una forma muy genérica y puede referirse a un “bloque de transformada (TB), un bloque de predicción (PB) o un bloque de unidad de codificación (CU)”. El filtro de pulsación más prolongada para los componentes de luma modifica un máximo de 7 muestras por cada línea de muestras perpendicular y adyacente al borde y se aplica a los bloques cuyo tamaño es ≥ 32 muestras en la dirección del filtro de desbloqueo, es decir, para los bordes verticales, la anchura del bloque debe ser ≥ 32 muestras y para los bordes horizontales, la altura del bloque debe ser ≥ 32 muestras.

El filtro de pulsación más prolongada para los componentes de croma se aplica a los bloques de croma cuando ambos bloques de croma adyacentes a un borde dado tienen un tamaño ≥ 8 muestras y modifica un máximo de tres muestras a cada lado del borde. Por lo tanto, para los bordes verticales, la anchura de bloque de ambos bloques de croma adyacentes al borde debe ser ≥ 8 muestras y para los bordes horizontales, la altura de bloque de ambos bloques adyacentes al borde debe ser ≥ 8 muestras.

El texto de especificación de filtro de desbloqueo en la VVC 6.0 se adjunta a continuación:

Proceso de filtro de desbloqueo

8.8.3.1 Generalidades

Las entradas de este proceso son la imagen reconstruida antes del filtro de desbloqueo, es decir, la matriz $recPicture_L$ y, cuando $ChromaArrayType$ no es igual a 0, las matrices $recPicture_{Cb}$ y $recPicture_{Cr}$.

Las salidas de este proceso son la imagen reconstruida modificada después de desbloquear, es decir, la matriz $recPicture_L$ y, cuando $ChromaArrayType$ no es igual a 0, las matrices $recPicture_{Cb}$ y $recPicture_{Cr}$.

Los bordes verticales de una imagen se filtran primero. A continuación, los bordes horizontales de una imagen se filtran con muestras modificadas mediante el proceso de filtrado de bordes verticales como entrada. Los bordes verticales y horizontales de los CTB de cada CTU se procesan por separado basándose en una unidad de codificación. Los bordes verticales de los bloques de codificación en una unidad de codificación se filtran comenzando por el borde del lado izquierdo de los bloques de codificación avanzando a través de los bordes hacia el lado derecho de los bloques de codificación en su orden geométrico. Los bordes horizontales de los bloques de codificación en una unidad de codificación se filtran comenzando por el borde en la parte superior de los bloques de codificación avanzando a través de los bordes hacia la parte inferior de los bloques de codificación en su orden geométrico.

NOTA: aunque el proceso de filtrado se especifica por imagen en esta especificación, el proceso de filtrado puede implementarse por unidad de codificación con un resultado equivalente, siempre que el decodificador tenga en cuenta adecuadamente el orden de dependencia de procesamiento para producir los mismos valores de salida.

El proceso de filtro de desbloqueo se aplica a todos los bordes de los subbloques de codificación y a los bordes de los bloques de transformada de una imagen, excepto a los siguientes tipos de bordes:

- los bordes que están en el límite de la imagen,
- los bordes que coinciden con los límites de una subimagen para la que $loop_filter_across_subpic_enabled_flag$ [SubPicIdx] es igual a 0,
- los bordes que coinciden con los límites virtuales de la imagen cuando $pps_loop_filter_across_virtual_boundaries_disabled_flag$ es igual a 1,
- los bordes que coinciden con los límites de ladrillo cuando $loop_filter_across_bricks_enabled_flag$ es igual a 0,

- los bordes que coinciden con los límites de segmento cuando `loop_filter_across_slices_enabled_flag` es igual a 0,
- los bordes que coinciden con los límites superiores o izquierdos de los segmentos con `slice_deblocking_filter_disabled_flag` igual a 1,
- los bordes dentro de segmentos con `slice_deblocking_filter_disabled_flag` igual a 1,
- los bordes que no corresponden a los límites de la cuadrícula de muestra de 4 x 4 del componente de luma,
- los bordes que no corresponden a los límites de la cuadrícula de muestra de 8x8 del componente de croma,
- los bordes dentro del componente de luma para los cuales ambos lados del borde tienen `intra_bdpcm_flag` igual a 1,
- los bordes de subbloques de croma que no son bordes de la unidad de transformada asociada.

El tipo de borde, vertical u horizontal, se representa mediante la variable `edgeType`, tal como se especifica en la Tabla 8-17.

Tabla 8-17: Nombre de la asociación a `edgeType`

Tipo de borde	Nombre del tipo de borde
0 (borde vertical)	EDGE_VER
1 (borde horizontal)	EDGE_HOR

Cuando `slice_deblocking_filter_disabled_flag` del segmento actual es igual a 0, se aplica lo siguiente:

- la variable `treeType` se establece igual a `DUAL_TREE_LUMA`.
- Los bordes verticales se filtran invocando al proceso de filtro de desbloqueo para una dirección tal como se especifica en la cláusula 8.8.3.2 con la variable `treeType`, la imagen reconstruida antes de desbloquear, es decir, la matriz `recPictureL`, y la variable `edgeType` establecidas igual a `EDGE_VER` como entradas, y la imagen reconstruida modificada después de desbloquear, es decir, la matriz `recPictureL` como salidas.
- El borde horizontal se filtra invocando al proceso de filtro de desbloqueo para una dirección tal como se especifica en la cláusula 8.8.3.2 con la variable `treeType`, la imagen reconstruida modificada después de desbloquear, es decir, la matriz `recPictureL`, y la variable `edgeType` establecidas igual a `EDGE_HOR` como entradas, y la imagen reconstruida modificada después de desbloquear, es decir, la matriz `recPictureL` como salidas.
- Cuando `ChromaArrayType` no es igual a 0, se aplica lo siguiente:
 - la variable `treeType` se establece igual a `DUAL_TREE_CHROMA`
 - los bordes verticales se filtran invocando al proceso de filtro de desbloqueo para una dirección tal como se especifica en la cláusula 8.8.3.2 con la variable `treeType`, la imagen reconstruida antes de desbloquear, es decir, las matrices `recPictureCb` y `recPictureCr`, y la variable `edgeType` establecidas igual a `EDGE_VER` como entradas, y la imagen reconstruida modificada después de desbloquear, es decir, las matrices `recPictureCb` y `recPictureCr` como salidas.
 - El borde horizontal se filtra invocando al proceso de filtro de desbloqueo para una dirección tal como se especifica en la cláusula 8.8.3.2 con la variable `treeType`, la imagen reconstruida modificada después de desbloquear, es decir, las matrices `recPictureCb` y `recPictureCr`, y la variable `edgeType` establecida igual a `EDGE_HOR` como entradas, y la imagen reconstruida modificada después de desbloquear, es decir, las matrices `recPictureCb` y `recPictureCr` como salidas.

8.8.3.2 Proceso de filtro de desbloqueo para una dirección

Las entradas de este proceso son:

- la variable `treeType` que especifica si los componentes de luma (`DUAL_TREE_LUMA`) o croma (`DUAL_TREE_CHROMA`) se están procesando actualmente,
- cuando `treeType` es igual a `DUAL_TREE_LUMA`, la imagen reconstruida antes de desbloquear, es decir, la matriz `recPictureL`,

– cuando ChromaArrayType no es igual a 0 y treeType es igual a DUAL_TREE_CHROMA, las matrices $recPicture_{Cb}$ y $treeType_{Cr}$,

5 – una variable edgeType que especifica si se filtra un borde vertical (EDGE_VER) u horizontal (EDGE_HOR).

Los resultados de este proceso son la imagen reconstruida modificada después de desbloquear, es decir:

– cuando treeType es igual a DUAL_TREE_LUMA, la matriz $recPicture_L$,

10 – cuando ChromaArrayType no es igual a 0 y treeType es igual a DUAL_TREE_CHROMA, las matrices $recPicture_{Cb}$ y $recPicture_{Cr}$.

Las variables firstCompldx and lastCompldx se derivan del siguiente modo:

15 $firstCompldx = (treeType == DUAL_TREE_CHROMA) ? 1 : 0$ (8-1022)

$lastCompldx = (treeType == DUAL_TREE_LUMA || ChromaArrayType == 0) ? 0 : 2(8-1023)$

20 Para cada unidad de codificación y cada bloque de codificación por componente de color de una unidad de codificación indicada por el índice de componente de color cldx que va desde firstCompldx hasta lastCompldx, ambos inclusive, con la anchura del bloque de codificación nCbW, la altura del bloque de codificación nCbH y la ubicación de la muestra superior izquierda del bloque de codificación (xCb, yCb), cuando cldx es igual a 0, o cuando cldx no es igual a 0 y edgeType es igual a EDGE_Type VER y $xCb \% 8$ es igual a 0, o cuando cldx no es igual a 0 y edgeType es igual a EDGE_HOR e $yCb \% 8$ es igual a 0, los bordes se filtran siguiendo las siguientes etapas ordenadas:

25 1. La variable filterEdgeFlag se deriva del siguiente modo:

– Si edgeType es igual a EDGE_VER y se cumplen una o más de las siguientes condiciones, filterEdgeFlag se establece igual a 0:

30 – El límite izquierdo del bloque de codificación actual es el límite izquierdo de la imagen.

– El límite izquierdo del bloque de codificación actual es el límite izquierdo o derecho de la subimagen y $loop_filter_across_subpic_enabled_flag$ [SubPicldx] es igual a 0.

35 – El límite izquierdo del bloque de codificación actual es el límite izquierdo del ladrillo y $loop_filter_across_bricks_enabled_flag$ es igual a 0.

40 – El límite izquierdo del bloque de codificación actual es el límite izquierdo del segmento y $loop_filter_across_bricks_enabled_flag$ es igual a 0.

– El límite izquierdo del bloque de codificación actual es uno de los límites virtuales verticales de la imagen y $pps_loop_filter_across_virtual_boundaries_disabled_flages$ igual a 1.

45 – De lo contrario, si edgeType es igual a EDGE_HOR y se cumplen una o más de las siguientes condiciones, la variable filterEdgeFlag se establece igual a 0:

– El límite superior del bloque de codificación de luma actual es el límite superior de la imagen.

50 – El límite superior del bloque de codificación actual es el límite superior o inferior de la subimagen y $loop_filter_across_subpic_enabled_flag$ [SubPicldx] es igual a 0.

– El límite superior del bloque de codificación actual es el límite superior del ladrillo y $loop_filter_across_bricks_enabled_flag$ es igual a 0.

55 – El límite superior del bloque de codificación actual es el límite superior del segmento y $loop_filter_across_slices_enabled_flag$ es igual a 0.

60 – El límite superior del bloque de codificación actual es uno de los límites virtuales horizontales de la imagen y $pps_loop_filter_across_virtual_boundaries_disabled_flages$ igual a 1.

– De lo contrario, filterEdgeFlag se establece igual a 1.

65 2. Todos los elementos de la matriz bidimensional $(nCbW) \times (nCbH)$ edgeFlags, maxFilterLengthQs y maxFilterlengthPs se inicializan para ser iguales a cero.

3. El proceso de derivación del límite del bloque de transformada especificado en la cláusula 8.8.3.3 se invoca con la ubicación (xCb, yCb), la anchura del bloque de codificación nCbW, la altura del bloque de codificación nCbH, la variable cldx, la variable filterEdgeFlag, la matriz edgeFlags, las matrices de longitud máxima de filtro maxFilterLengthP y maxFilterLengthQs y la variable edgeType como entradas, y la matriz modificada edgeFlags, las matrices de longitud máxima de filtro modificada maxFilterLengthPs y maxFilterLengthQs como salidas.

4. Cuando cldx es igual a 0, se invoca al proceso de derivación del límite del subbloque de codificación especificado en la cláusula 8.8.3.4 con la ubicación (xCb, yCb), la anchura del bloque de codificación nCbW, la altura del bloque de codificación nCbH, la matriz edgeFlags, las matrices de longitud máxima de filtro maxFilterLengthPs y maxFilterLengthQs y la variable edgeType como entradas, y la matriz modificada edgeFlags, las matrices de longitud máxima de filtro modificada maxFilterLengthP y maxFilterLengthQs como salidas.

5. La matriz de muestra de imagen recPicture se obtiene del siguiente modo:

- Si cldx es igual a 0, recPicture se establece igual a la matriz de muestra de imagen de luma reconstruida antes de desbloquear recPicture_L.

- De lo contrario, si cldx es igual a 1, recPicture se establece igual a la matriz de muestra de imagen de croma reconstruida antes de desbloquear recPicture_{CB}.

- De lo contrario (cldx es igual a 2), recPicture se establece igual a la matriz de muestra de imagen de croma reconstruida antes de desbloquear recPicture_{Cr}.

6. El proceso de derivación de la intensidad de filtrado de límites especificado en la cláusula 8.8.3.5 se invoca con la matriz de muestra de imagen recPicture, la ubicación de luma (xCb, yCb), la anchura del bloque de codificación nCbW, la altura del bloque de codificación nCbH, la variable edgeType, la variable cldx y la matriz edgeFlags como entradas y una matriz bS (nCbW)x(nCbH) como salida.

7. El proceso de filtrado de bordes para una dirección se invoca para un bloque de codificación tal como se especifica en la cláusula 8.8.3.6 con la variable edgeType, la variable cldx, la imagen reconstruida antes de desbloquear recPicture, la ubicación (xCb, yCb), la anchura del bloque de codificación nCbW, la altura del bloque de codificación nCbH y las matrices bS, maxFilterLengthPs y maxFilterLengthQs, como entradas, y la imagen reconstruida modificada recPicture como salida.

8.8.3.3 Proceso de derivación del límite del bloque de transformada

Las entradas de este proceso son:

- una ubicación (xCb, yCb) que especifica la muestra superior izquierda del bloque de codificación actual en relación con la muestra superior izquierda de la imagen actual,
- una variable nCbW que especifica la anchura del bloque de codificación actual,
- una variable nCbH que especifica la altura del bloque de codificación actual,
- una variable cldx que especifica el componente de color del bloque de codificación actual,
- una variable filterEdgeFlag,
- una matriz bidimensional (nCbW)x(nCbH) edgeFlags,
- matrices bidimensionales (nCbW)x(nCbH) maxFilterLengthQs y maxFilterLengthPs,
- una variable edgeType que especifica si se filtra un borde vertical (EDGE_VER) u horizontal (EDGE_HOR).

Las salidas de este proceso son:

- la matriz bidimensional (nCbW)x(nCbH) modificada edgeFlags,
- las matrices bidimensionales (nCbW)x(nCbH) modificadas maxFilterLengthQs, maxFilterLengthPs.

Dependiendo de EdgeType, las matrices edgeFlags, maxFilterLengthPs y maxFilterLengthQs se derivan del siguiente modo:

- la variable gridSize se establece del siguiente modo:

$$\text{gridSize} = \text{cldx} = 0? 4: 8(8-1024)$$

- Si edgeType es igual a EDGE_VER, se aplica lo siguiente:
- 5 - la variable numEdges se establece igual a $\text{Max}(1, \text{nCbW} / \text{gridSize})$.
- Para xEdge = 0.. numEdges - 1 e y = 0.. nCbH - 1, se aplica lo siguiente:
- la posición horizontal x dentro del bloque de codificación actual se establece igual a $\text{xEdge} * \text{gridSize}$.
- 10 - El valor de edgeFlags[x][y] se obtiene del siguiente modo:
- Si pps_loop_filter_across_virtual_boundaries_disabled_flag es igual a 1 y $(\text{xCb} + \text{x})$ es igual a PpsVirtualBoundariesPosX [n] para cualquier $n = 0.. \text{pps_num_ver_virtual_boundaries} - 1$, edgeFlags[x][y] se establece igual a 0.
- 15 - De lo contrario, si x es igual a 0, edgeFlags [x] [y] se establece igual a filterEdgeFlag.
- De lo contrario, si la ubicación $(\text{xCb} + \text{x}, \text{yCb} + \text{y})$ está en el borde de un bloque de transformada, edgeFlags [x] [y] se establece igual a 1.
- 20 - Cuando edgeFlags[x][y] es igual a 1, se aplica lo siguiente:
- Si cldx es igual a 0, se aplica lo siguiente:
- 25 - el valor de maxFilterLengthQs [x][y] se obtiene del siguiente modo:
- Si la anchura en las muestras de luma del bloque de transformada en la ubicación de luma $(\text{xCb} + \text{x}, \text{yCb} + \text{y})$ es igual o menor que 4 o la anchura en las muestras de luma del bloque de transformada en la ubicación de luma $(\text{xCb} + \text{x} - 1, \text{yCb} + \text{y})$ es igual o menor que 4, maxFilterLengthQs [x][y] se establece igual a 1.
- 30 - De lo contrario, si la anchura en las muestras de luma del bloque de transformada en la ubicación de luma $(\text{xCb} + \text{x}, \text{yCb} + \text{y})$ es igual o mayor que 32, maxFilterLengthQs [x] [y] se establece igual a 7.
- De lo contrario, maxFilterLengthQs [x] [y] se establece igual a 3.
- 35 - El valor de maxFilterLengthPs [x] [y] se obtiene del siguiente modo:
- Si la anchura en las muestras de luma del bloque de transformada en la ubicación de luma $(\text{xCb} + \text{x}, \text{yCb} + \text{y})$ es igual o menor que 4 o la anchura en las muestras de luma del bloque de transformada en la ubicación de luma $(\text{xCb} + \text{x} - 1, \text{yCb} + \text{y})$ es igual o menor que 4, maxFilterLengthPs [x] [y] se establece igual a 1.
- 40 - De lo contrario, si la anchura en las muestras de luma del bloque de transformada en la ubicación de luma $(\text{xCb} + \text{x} - 1, \text{yCb} + \text{y})$ es igual o mayor que 32, maxFilterLengthPs [x] [y] se establece igual a 7.
- 45 - De lo contrario, maxFilterLengthPs [x] [y] se establece igual a 3.
- De lo contrario (cidX no es igual a 0), los valores de maxFilterLengthPs [x] [y] y maxFilterLengthQs [x] [y] se derivan del siguiente modo:
- 50 - Si la anchura en las muestras de croma del bloque de transformada en la ubicación de croma $(\text{xCb} + \text{x}, \text{yCb} + \text{y})$ y la anchura en la ubicación de croma $(\text{xCb} + \text{x} - 1, \text{yCb} + \text{y})$ son iguales o superiores a 8, maxFilterLengthPs [x] [y] y maxFilterLengthQs [x] [y] se establecen en 3.
- De lo contrario, maxFilterLengthPs [x] [y] y maxFilterLengthQs [x] [y] se establecen en 1.
- 55 - De lo contrario (edgeType es igual a EDGE_HOR), se aplica lo siguiente:
- La variable numEdges se establece igual a $\text{Max}(1, \text{nCbH} / \text{gridSize})$.
- 60 - Para yEdge = 0.. numEdges - 1 y x = 0.. nCbW - 1, se aplica lo siguiente:
- La posición vertical y dentro del bloque de codificación actual se establece igual a $\text{yEdge} * \text{gridSize}$.
- El valor de edgeFlags[x][y] se obtiene del siguiente modo:
- 65

- Si `pps_loop_filter_across_virtual_boundaries_disabled_flag` es igual a 1 y $(yCb + y)$ es igual a `PpsVirtualBoundariesPosY [n]` para cualquier $n = 0..pps_num_hor_virtual_boundaries - 1$, `edgeFlags [x] [y]` se establece igual a 0.
- 5 - De lo contrario, si y es igual a 0, `edgeFlags [x] [y]` se establece igual a `filterEdgeFlag`.
- De lo contrario, si la ubicación $(xCb + x, yCb + y)$ está en el borde de un bloque de transformada, `edgeFlags [x] [y]` se establece igual a 1.
- 10 - Cuando `edgeFlags[x][y]` es igual a 1, se aplica lo siguiente:
 - Si `cldx` es igual a 0, se aplica lo siguiente:
 - el valor de `maxFilterLengthQs [x][y]` se obtiene del siguiente modo:
 - 15 - Si la altura en las muestras de luma del bloque de transformada en la ubicación de luma $(xCb + x, yCb + y)$ es igual o menor que 4 o la altura en las muestras de luma del bloque de transformada en la ubicación de luma $(xCb + x, yCb + y - 1)$ es igual o menor que 4, `maxFilterLengthQs [x] [y]` se establece igual a 1.
 - 20 - De lo contrario, si la altura en las muestras de luma del bloque de transformada en la ubicación de luma $(xCb + x, yCb + y)$ es igual o mayor que 32, `maxFilterLengthQs [x] [y]` se establece igual a 7.
 - De lo contrario, `maxFilterLengthQs [x] [y]` se establece igual a 3.
 - 25 - El valor de `maxFilterLengthPs [x] [y]` se obtiene del siguiente modo:
 - Si la altura en las muestras de luma del bloque de transformada en la ubicación de luma $(xCb + x, yCb + y)$ es igual o menor que 4 o la altura en las muestras de luma del bloque de transformada en la ubicación de luma $(xCb + x, yCb + y - 1)$ es igual o menor que 4, `maxFilterLengthps [x] [y]` se establece igual a 1.
 - 30 - De lo contrario, si la altura en las muestras de luma del bloque de transformada en la ubicación de luma $(xCb + x, yCb + y - 1)$ es igual o mayor que 32, `maxFilterLengthPs [x] [y]` se establece igual a 7.
 - De lo contrario, `maxFilterLengthPs [x] [y]` se establece igual a 3.
 - 35 - De lo contrario (`cidX` no es igual a 0), los valores de `maxFilterLengthPs [x] [y]` y `maxFilterLengthQs [x] [y]` se derivan del siguiente modo:
 - Si se cumplen todas las condiciones siguientes, `maxFilterLengthPs [x] [y]` y `maxFilterLengthQs [x] [y]` se establecen igual a 3:
 - la altura en las muestras de croma del bloque de transformada en la ubicación de croma $(xCb + x, yCb + y)$ y la altura en la ubicación de croma $(xCb + x, yCb + y - 1)$ son iguales o mayores que 8.
 - 45 - $(yCb + y) \% CtbHeightC$ es mayor que 0, es decir, el borde horizontal no se superpone con el límite superior del CTB de croma.
 - De lo contrario, `maxFilterLengthPs [x] [y]` y `maxFilterLengthQs [x] [y]` se establecen igual a 1.
- 50 8.8.3.4 Proceso de derivación del límite del subbloque de codificación
- Las entradas de este proceso son:
 - una ubicación (xCb, yCb) que especifica la muestra superior izquierda del bloque de codificación actual en relación con la muestra superior izquierda de la imagen actual,
 - 55 - una variable `nCbW` que especifica la anchura del bloque de codificación actual,
 - una variable `nCbH` que especifica la altura del bloque de codificación actual,
 - 60 - una matriz bidimensional $(nCbW) \times (nCbH)$ `edgeFlags`,
 - matrices bidimensionales $(nCbW) \times (nCbH)$ `maxFilterLengthQs` y `maxFilterLengthPs`,
 - 65 - una variable `edgeType` que especifica si se filtra un borde vertical (`EDGE_VER`) u horizontal (`EDGE_HOR`).

Las salidas de este proceso son:

- la matriz bidimensional $(nCbW) \times (nCbH)$ modificada `edgeFlags`,
- 5 - las matrices bidimensionales $(nCbW) \times (nCbH)$ modificadas `maxFilterLengthQs` y `maxFilterLengthPs`.

El número de subbloques de codificación en la dirección horizontal `numSbX` y en la dirección vertical `numSbY` se obtiene del siguiente modo:

- 10 - Si `inter_affine_flag [xCb] [yCb]` es igual a 1 o `merge_subblock_flag [xCb] [yCb]` es igual a 1, `numSbX` y `numSbY` se establecen iguales a `NumSbX [xCb] [yCb]` y `NumSbY [xCb] [yCb]`, respectivamente.
- De lo contrario, `numSbX` y `numSbY` se establecen iguales a 1.

15 Dependiendo del valor de `edgeType`, se aplica lo siguiente:

- Si `edgeType` es igual a `EDGE_VER`, se aplica lo siguiente:
- 20 - la variable `sbW` se establece igual a $\text{Max}(8, nCbW/\text{numSbX})$.
- La matriz `edgeTbFlags` se establece igual a `edgeFlags`.
- Para $xEdge = 0.. \min((nCbW/8) - 1, \text{numSbX} - 1)$, $y = 0.. nCbH - 1$:
- 25 - La posición horizontal `x` dentro del bloque de codificación actual se establece igual a $xEdge * sbW$.
- El valor de `edgeFlags [x] [y]` se deriva del siguiente modo:
- 30 - Si `pps_loop_filter_across_virtual_boundaries_disabled_flag` es igual a 1 y `x` es igual a `PpsVirtualBoundariesPosX [n]` para cualquier $n = 0..pps_num_ver_virtual_boundaries - 1$, se aplica lo siguiente:

$$\text{edgeFlags}[x] [y] = 0 \quad (8-1025)$$

- 35 - De lo contrario, se aplica lo siguiente:

$$\text{edgeFlags} [x] [y] = 1 \quad (8-1026)$$

- Cuando `edgeFlags [x] [y]` es igual a 1, los valores de `maxFilterLengthPs [x] [y]` y `maxFilterLengthQs [x] [y]` se modifican del siguiente modo:

- Si `x` es igual a 0, se aplica lo siguiente:
- Cuando `numSbX` es mayor que 1, se aplica lo siguiente:

$$45 \quad \text{maxFilterLengthQs}[x][y] = \text{Min}(5, \text{maxFilterLengthQs}[x][y]) \quad (8-1027)$$

- Cuando `inter_affine_flag [xCb - 1] [yCb]` es igual a 1 o `merge_subblock_flag [xCb - 1] [yCb]` es igual a 1, se aplica lo siguiente:

$$50 \quad \text{maxFilterLengthPs}[x][y] = \text{Min}(5, \text{maxFilterLengthPs}[x][y]) (8-1028)$$

- De lo contrario, si `edgeTbFlags [x] [y]` es igual a 1, se aplica lo siguiente:

$$55 \quad \text{maxFilterLengthPs}[x][y] = \text{Min}(5, \text{maxFilterLengthPs}[x][y]) (8-1029)$$

$$\text{maxFilterLengthQs}[x][y] = \text{Min}(5, \text{maxFilterLengthQs}[x][y]) (8-1030)$$

- De lo contrario, si se cumplen una o más de las siguientes condiciones:

- 60 - $(x + 4)$ es mayor o igual a `nCbW`,
- `edgeTbFlags [x - 4] [y]` es igual a 1,
- 65 - `edgeTbFlags [x + 4] [y]` es igual a 1,

se aplica lo siguiente:

$$\text{maxFilterLengthPs}[x][y] = 1 \quad (8-1031)$$

5 $\text{maxFilterLengthQs}[x][y] = 1 \quad (8-1032)$

- De lo contrario, si se cumplen una o más de las siguientes condiciones:

- xEdge es igual a 1,

10 - xEdge es igual a $(nCbW/8) - 1$,

- edgeTbFlags $[x - sbW][y]$ es igual a 1,

15 - edgeTbFlags $[x + sbW][y]$ es igual a 1, se aplica lo siguiente:

$$\text{maxFilterLengthPs}[x][y] = 2 \quad (8-1033)$$

20 $\text{maxFilterLengthQs}[x][y] = 2 \quad (8-1034)$

- De lo contrario, se aplica lo siguiente:

$$\text{maxFilterLengthPs}[x][y] = 3 \quad (8-1035)$$

25 $\text{maxFilterLengthQs}[x][y] = 3 \quad (8-1036)$

- De lo contrario, si edgeType es igual a EDGE_HOR, se aplica lo siguiente:

- la variable sbH se establece igual a $\text{Max}(8, nCbH/\text{numSbY})$.

30 - La matriz edgeTbFlags se establece igual a edgeFlags.

- Para $yEdge = 0..\text{min}((nCbH/8) - 1, \text{numSbY} - 1)$, $x = 0..nCbW - 1$:

35 - La posición vertical y dentro del bloque de codificación actual se establece igual a $yEdge * sbH$.

- El valor de edgeFlags $[x][y]$ se deriva del siguiente modo:

40 - Si `pps_loop_filter_across_virtual_boundaries_disabled_flag` es igual a 1 e y es igual a `PpsVirtualBoundariesPosY[n]` para cualquier $n = 0..\text{pps_num_hor_virtual_boundaries} - 1$, se aplica lo siguiente:

$$\text{edgeFlags}[x][y] = 0 \quad (8-1037)$$

- De lo contrario, se aplica lo siguiente:

45 $\text{edgeFlags}[x][y] = 1 \quad (8-1038)$

- Cuando $\text{edgeFlags}[x][y]$ es igual a 1, los valores de $\text{maxFilterLengthPs}[x][y]$ y $\text{maxFilterLengthQs}[x][y]$ se modifican del siguiente modo:

50 - Si y es igual a 0 y $\text{edgeFlags}[x][y]$ es igual a 1, se aplica lo siguiente:

- Cuando numSbY es mayor que 1, se aplica lo siguiente:

55 $\text{maxFilterLengthQs}[x][y] = \text{Min}(5, \text{maxFilterLengthQs}[x][y]) \quad (81039)$

- Cuando $\text{inter_affine_flag}[xCb][yCb - 1]$ es igual a 1 o $\text{merge_subblock_flag}[xCb][yCb - 1]$ es igual a 1, se aplica lo siguiente:

60 $\text{maxFilterLengthPs}[x][y] = \text{Min}(5, \text{maxFilterLengthPs}[x][y]) \quad (8-1040)$

- De lo contrario, si $\text{edgeTbFlags}[x][y]$ es igual a 1, se aplica lo siguiente:

65 $\text{maxFilterLengthPs}[x][y] = \text{Min}(5, \text{maxFilterLengthPs}[x][y]) \quad (8-1041)$

$$\text{maxFilterLengthQs}[x][y] = \text{Min}(5, \text{maxFilterLengthQs}[x][y]) \quad (8-1042)$$

- De lo contrario, si se cumplen una o más de las siguientes condiciones:

- $(y + 4)$ es mayor o igual a $nCbH$,

5 - $edgeTbFlags[x][y - 4]$ es igual a 1,

- $edgeTbFlags[x][y + 4]$ es igual a 1, se aplica lo siguiente:

10
$$maxFilterLengthPs[x][y] = 1 \quad (8-1045)$$

$$maxFilterLengthQs[x][y] = 1 \quad (8-1046)$$

- De lo contrario, si se cumplen una o más de las siguientes condiciones:

15 - $yEdge$ es igual a 1,

- $yEdge$ es igual a $(nCbH/8) - 1$,

20 - $edgeTbFlags[x][y - sbH]$ es igual a 1,

- $edgeTbFlags[x][y + sbH]$ es igual a 1, se aplica lo siguiente:

25
$$maxFilterLengthPs[x][y] = 2 \quad (8-1043)$$

$$maxFilterLengthQs[x][y] = 2 \quad (8-1044)$$

- De lo contrario, se aplica lo siguiente:

30
$$maxFilterLengthPs[x][y] = 3 \quad (8-1047)$$

$$maxFilterLengthQs[x][y] = 3 \quad (8-1048)$$

8.8.3.5 Proceso de derivación de la intensidad de filtrado de límites

35 Las entradas de este proceso son:

- una matriz de muestra de imagen $recPicture$,

40 - una ubicación (xCb, yCb) que especifica la muestra superior izquierda del bloque de codificación actual en relación con la muestra superior izquierda de la imagen actual,

- una variable $nCbW$ que especifica la anchura del bloque de codificación actual,

45 - una variable $nCbH$ que especifica la altura del bloque de codificación actual,

- una variable $edgeType$ que especifica si se filtra un borde vertical (EDGE_VER) u horizontal (EDGE_HOR),

- una variable $cldx$ que especifica el componente de color del bloque de codificación actual,

50 - una matriz bidimensional $(nCbW) \times (nCbH)$ $edgeFlags$.

La salida de este proceso es una matriz bidimensional $(nCbW) \times (nCbH)$ bS que especifica la intensidad de filtrado de límites.

55 Las variables xDi , yDj , xN e yN se derivan del siguiente modo:

- la variable $gridSize$ se establece del siguiente modo:

$$gridSize = cldx = 0 ? 4 : 8 \quad (8-1049)$$

60 - Si $edgeType$ es igual a EDGE_VER,

$$xDi = (i * gridSize) \quad (8-1050)$$

65
$$yDj = cldx = 0 ? (j << 2) : (j << 1) \quad (8-1051)$$

$$xN \text{ se establece igual a } \text{Max}(0, (\text{nCbW} / \text{gridSize}) - 1) \quad (8-1052)$$

$$yN = \text{cldx} == 0 ? (\text{nCbH} / 4) - 1 : (\text{nCbH} / 2) - 1 \quad (8-1053)$$

5 – De lo contrario (edgeType es igual a EDGE_HOR),

$$xD_i = \text{cldx} == 0 ? (i << 2) : (i << 1) \quad (8-1054)$$

$$yD_j = (j * \text{gridSize}) \quad (8-1055)$$

$$10 \quad xN = \text{cldx} == 0 ? (\text{nCbW} / 4) - 1 : (\text{nCbW} / 2) - 1 \quad (8-1056)$$

$$yN = \text{Max}(0, (\text{nCbH} / \text{gridSize}) - 1) \quad (8-1057)$$

15 Para xD_i con $i = 0.. xN$ y yD_j con $j = 0.. yN$, se aplica lo siguiente:

– Si edgeFlags [xD_i] [yD_j] es igual a 0, la variable $bS[xD_i][yD_j]$ se establece igual a 0.

20 – De lo contrario, se aplica lo siguiente:

- los valores de muestra p_0 y q_0 se derivan del siguiente modo:

25 - si edgeType es igual a EDGE_VER, p_0 se establece igual a $\text{recPicture}[xCb + xD_i - 1][yCb + yD_j]$ y q_0 se establece igual a $\text{recPicture}[xCb + xD_i][yCb + yD_j]$.

- De lo contrario (edgeType es igual a EDGE_HOR), p_0 se establece igual a $\text{recPicture}[xCb + xD_i][yCb + yD_j - 1]$ y q_0 se establece igual a $\text{recPicture}[xCb + xD_i][yCb + yD_j]$.

30 - La variable $bS[xD_i][yD_j]$ se deriva del siguiente modo:

- Si cldx es igual a 0 y ambas muestras p_0 y q_0 están en un bloque de codificación con intra_bdpcm_flag igual a 1, $bS[xD_i][yD_j]$ se establece igual a 0.

35 - De lo contrario, si la muestra p_0 o q_0 está en el bloque de codificación de una unidad de codificación codificada con el modo de intrapredicción, $bS[xD_i][yD_j]$ se establece igual a 2.

- De lo contrario, si el borde del bloque también es un borde del bloque de transformada y la muestra p_0 o q_0 está en un bloque de codificación con ciip_flag igual a 1, $bS[xD_i][yD_j]$ se establece igual a 2.

40 - De lo contrario, si el borde del bloque también es un borde del bloque de transformada y la muestra p_0 o q_0 está en un bloque de transformada que contiene uno o más niveles de coeficientes de transformada distintos de cero, $bS[xD_i][yD_j]$ se establece igual a 1.

45 - De lo contrario, si el borde del bloque es también un borde del bloque de transformada, cldx es mayor que 0, y la muestra p_0 o q_0 está en una unidad de transformada con $\text{tu_joint_cbcr_residual_flag}$ igual a 1, $bS[xD_i][yD_j]$ se establece igual a 1.

50 - De lo contrario, si el modo de predicción del subbloque de codificación que contiene la muestra p_0 es diferente del modo de predicción del subbloque de codificación que contiene la muestra q_0 (es decir, uno de los subbloques de codificación se codifica en el modo de predicción IBC y el otro se codifica en el modo de interpredicción), $bS[xD_i][yD_j]$ se establece igual a 1.

55 - De lo contrario, si cldx es igual a 0 y se cumplen una o más de las siguientes condiciones, $bS[xD_i][yD_j]$ se establece igual a 1:

- El subbloque de codificación que contiene la muestra p_0 y el subbloque de codificación que contiene la muestra q_0 se codifican ambos en el modo de predicción IBC, y la diferencia absoluta entre el componente horizontal o vertical de los vectores de bloque usados en la predicción de los dos subbloques de codificación es mayor o igual a 8 en unidades de 1/16 de muestras de luma.

60 - Para la predicción del subbloque de codificación que contiene la muestra p_0 se utilizan diferentes imágenes de referencia o un número diferente de vectores de movimiento que para la predicción del subbloque de codificación que contiene la muestra q_0 .

65 NOTA 1: la determinación de si las imágenes de referencia utilizadas para los dos subbloques de codificación son las mismas o diferentes se basa únicamente en las imágenes a las que se hace referencia, sin tener en cuenta si se forma una predicción

utilizando un índice en la lista 0 de imágenes de referencia o un índice en la lista 1 de imágenes de referencia, y también sin tener en cuenta si la posición del índice dentro de una lista de imágenes de referencia es diferente.

5 NOTA 2: el número de vectores de movimiento que se utilizan para la predicción de un subbloque de codificación con una cobertura de muestra superior izquierda (x_{Sb}, y_{Sb}) es igual a $PredFlagL0[x_{Sb}][y_{Sb}] + PredFlagL1[x_{Sb}][y_{Sb}]$.

10 - Se usa un vector de movimiento para predecir el subbloque de codificación que contiene la muestra p_0 y se usa un vector de movimiento para predecir el subbloque de codificación que contiene la muestra q_0 , y la diferencia absoluta entre el componente horizontal o vertical de los vectores de movimiento utilizados es mayor o igual a 8 en unidades de 1/16 de muestras de luma.

15 - Se usan dos vectores de movimiento y dos imágenes de referencia diferentes para predecir el subbloque de codificación que contiene la muestra p_0 , dos vectores de movimiento para las mismas dos imágenes de referencia se usan para predecir el subbloque de codificación que contiene la muestra q_0 y la diferencia absoluta entre la componente horizontal o vertical de los dos vectores de movimiento utilizados en la predicción de los dos subbloques de codificación para la misma imagen de referencia es mayor o igual a 8 en unidades de 1/16 de muestras de luma.

20 - Se usan dos vectores de movimiento para la misma imagen de referencia para predecir el subbloque de codificación que contiene la muestra p_0 , dos vectores de movimiento para la misma imagen de referencia se usan para predecir el subbloque de codificación que contiene la muestra q_0 y las dos condiciones siguientes se cumplen:

25 - la diferencia absoluta entre el componente horizontal o vertical de los vectores de movimiento de la lista 0 utilizados en la predicción de los dos subbloques de codificación es mayor o igual a 8 en 1/16 muestras de luma, o la diferencia absoluta entre el componente horizontal o vertical de los vectores de movimiento de la lista 1 utilizados en la predicción de los dos subbloques de codificación es mayor o igual a 8 en unidades de 1/16 de muestras de luma.

30 - La diferencia absoluta entre el componente horizontal o vertical del vector de movimiento de la lista 0 utilizado en la predicción del subbloque de codificación que contiene la muestra p_0 y el vector de movimiento de la lista 1 utilizado en la predicción del subbloque de codificación que contiene la muestra q_0 es mayor o igual a 8 en unidades de 1/16 de muestras de luma, o la diferencia absoluta entre el componente horizontal o vertical del vector de movimiento de la lista 1 usado en la predicción del subbloque de codificación que contiene la muestra p_0 y el vector de movimiento de la lista 0 usado en la predicción del subbloque de codificación que contiene la muestra q_0 es mayor o igual a 8 en unidades de 1/16 de muestras de luma.

35 - De lo contrario, la variable $bS[xD_i][yD_i]$ se establece igual a 0.

8.8.3.6 Proceso de filtrado de bordes para una dirección

Las entradas de este proceso son:

- 40 - una variable `edgeType` que especifica si los bordes verticales (`EDGE_VER`) o los bordes horizontales (`EDGE_HOR`) se están procesando actualmente,
- 45 - una variable `cldx` que especifica el componente de color actual,
- la imagen reconstruida antes de desbloquear `recPicture`,
- 50 - una ubicación (x_{Cb}, y_{Cb}) que especifica la muestra superior izquierda del bloque de codificación actual en relación con la muestra superior izquierda de la imagen actual,
- una variable `nCbW` que especifica la anchura del bloque de codificación actual,
- una variable `nCbH` que especifica la altura del bloque de codificación actual,
- 55 - la matriz `bS` que especifica la intensidad de límite,
- las matrices `maxFilterLengthPs` y `maxFilterLengthQs`.

60 La salida de este proceso es la imagen reconstruida modificada después de desbloquear `recPicture`.

Para el proceso de filtrado de bordes, se aplica lo siguiente:

- 65 - la variable `gridSize` se establece del siguiente modo:

$$gridSize = cldx == 0 ? 4 : 8 \quad (8-1058)$$

–las variables subW, subH, xN, yN se derivan del siguiente modo:

$$\begin{aligned} \text{subW} &= \text{cldx} == 0 ? 1 : \text{SubWidthC} && (8-1059) \\ \text{subH} &= \text{cldx} == 0 ? 1 : \text{SubHeightC} && (8-1060) \\ \text{xN} &= \text{edgeType} == \text{EDGE_VER} ? \text{Max}(0, (\text{nCbW} / \text{gridSize}) - 1) : (\text{nCbW} / 4 / \text{subW}) - 1 && (8-1061) \\ \text{yN} &= \text{edgeType} == \text{EDGE_VER} ? (\text{nCbH} / 4 / \text{subH}) - 1 : \text{Max}(0, (\text{nCbH} / \text{gridSize}) - 1) && (8-1062) \end{aligned}$$

– las variables xD_k con k = 0..xN y yD_m con m = 0..yN se derivan del siguiente modo:

$$\begin{aligned} \text{xD}_k &= \text{edgeType} == \text{EDGE_VER} ? (k * \text{gridSize}) : (k << (2 / \text{subW})) && (8-1063) \\ \text{yD}_m &= \text{edgeType} == \text{EDGE_VER} ? (m << (2 / \text{subH})) : (m * \text{gridSize}) && (8-1064) \end{aligned}$$

– Para xD_k con k = 0..xN y yD_m con m = 0..yN, se aplica lo siguiente:

– Cuando bS[xD_k] [yD_m] es mayor que 0, se aplican las siguientes etapas ordenadas:

– si cldx es igual a 0, el proceso de filtrado para bordes en el bloque de codificación de luma de la unidad de codificación actual consiste en las siguientes etapas ordenadas:

1. El proceso de decisión para los bordes del bloque de luma, tal como se especifica en la cláusula 8.8.3.6.1, se invoca con la matriz de muestra de imagen de luma recPicture, la ubicación del bloque de codificación de luma (xCb, yCb), la ubicación de luma del bloque (xBl, yBl) establecida igual a (xD_k, yD_m), la dirección del borde edgeType, la intensidad de filtrado de límite bS[xD_k][yD_m], las longitudes de filtro máximas maxFilterLengthP establecidas igual a maxFilterLengthPs[xD_k][yD_m] y maxFilterLengthQ establecida igual a maxFilterLengthQs[xD_k][yD_m] como entradas, y las decisiones dE, dEp y dEq, las longitudes de filtro máximas modificadas maxFilterLengthP y maxFilterLengthQ, y la variable tc como salidas.

2. El proceso de filtrado para los bordes de bloque, tal como se especifica en la cláusula 8.8.3.6.2, se invoca con la matriz de muestra de imagen de luma recPicture, la ubicación del bloque de codificación de luma (xCb, yCb), la ubicación de luma del bloque (xBl, yBl) establecida igual a (xD_k, yD_m), la dirección de borde edgeType, las decisiones dE, dEp y dEq, las longitudes de filtro máximas maxFilterLengthP y maxFilterLengthQ y la variable tc como entradas, y la matriz de muestra de imagen de luma modificada recPicture como salida.

– De lo contrario (cldx no es igual a 0), el proceso de filtrado de los bordes en el bloque de codificación de croma de la unidad de codificación actual especificada por cldx consiste en las siguientes etapas ordenadas:

1. La variable cQpPicOffset se deriva del siguiente modo:

$$\text{cQpPicOffset} = \text{cldx} == 1 ? \text{pps_cb_qp_offset} : \text{pps_cr_qp_offset} \quad (8-1065)$$

2. El proceso de decisión para bordes de bloques de croma, tal como se especifica en la cláusula 8.8.3.6.3, se invoca con la matriz de muestra de imagen de croma recPicture, la ubicación del bloque de codificación de croma (xCb, yCb), la ubicación del bloque de croma (xBl, yBl) establecido igual a (xD_k, yD_m), la dirección de borde edgeType, la variable cldx, la variable cQpPicOffset, la intensidad de filtrado de límites bS[xD_k][yD_m], y la variable maxFilterLengthCbCr establecida igual a maxFilterLengthPs[xD_k][yD_m] como entradas, y la variable modificada maxFilterLengthCbCr y la variable tc como salidas.

3. Cuando maxFilterLengthCbCr es mayor que 0, el proceso de filtrado para bordes de bloques de croma, tal como se especifica en la cláusula 8.8.3.6.4, se invoca con la matriz de muestra de imagen de croma recPicture, la ubicación del bloque de codificación de croma (xCb, yCb), la ubicación de croma del bloque (xBl, yBl) establecida igual a (xD_k, yD_m), la dirección de borde edgeType, la variable maxFilterLengthCbCr y la variable tc como entradas, y la matriz de muestra de imagen de croma modificada recPicture como salida.

8.8.3.6.1 Proceso de decisión para bordes de bloque de luma

Las entradas de este proceso son:

– una matriz de muestra de imagen recPicture,

– una ubicación (xCb, yCb) que especifica la muestra superior izquierda del bloque de codificación actual en relación con la muestra superior izquierda de la imagen actual,

- una ubicación (xB1, yB1) que especifica la muestra superior izquierda del bloque actual en relación con la muestra superior izquierda del bloque de codificación actual,
- 5 – una variable edgeType que especifica si se filtra un borde vertical (EDGE_VER) u horizontal (EDGE_HOR),
- una variable bS que especifica la intensidad de filtrado de límites,
- una variable maxFilterLengthP que especifica la longitud máxima del filtro,
- 10 – una variable maxFilterLengthQ que especifica la longitud máxima del filtro.

Las salidas de este proceso son:

- 15 – las variables dE, dEp y dEq que contienen decisiones,
- las variables de longitud de filtro modificadas maxFilterLengthP y maxFilterLengthQ,
- 20 – la variable tc.

Los valores de muestra $p_{i,k}$ y $q_{j,k}$ con $i = 0.. \text{maxFilterLengthP}$, $j = 0.. \text{maxFilterLengthQ}$ y $k = 0$ y 3 se derivan del siguiente modo:

- 25 – Si edgeType es igual a EDGE_VER, se aplica lo siguiente:

$$q_{j,k} = \text{recPicture}_L[\text{xCb} + \text{xB1} + j][\text{yCb} + \text{yB1} + k] \quad (8-1066)$$

$$p_{i,k} = \text{recPicture}_L[\text{xCb} + \text{xB1} - i - 1][\text{yCb} + \text{yB1} + k] \quad (8-1067)$$

- 30 – De lo contrario (edgeType es igual a EDGE_HOR), se aplica lo siguiente:

$$q_{j,k} = \text{recPicture}[\text{xCb} + \text{xB1} + k][\text{yCb} + \text{yB1} + j] \quad (8-1068)$$

$$p_{i,k} = \text{recPicture}[\text{xCb} + \text{xB1} + k][\text{yCb} + \text{yB1} - i - 1] \quad (8-1069)$$

La variable qpOffset se deriva del siguiente modo:

- Si sps_ladf_enabled_flag es igual a 1, se aplica lo siguiente:

- la variable lumaLevel del nivel de luma reconstruido se obtiene del siguiente modo:

$$\text{lumaLevel} = ((p_{0,0} + p_{0,3} + q_{0,0} + q_{0,3}) \gg 2), \quad (8-1070)$$

- La variable qpOffset se establece igual a sps_ladf_lowest_interval_qp_offset y se modifica del siguiente modo:

para (i = 0); i < sps_num_ladf_intervals_minus2 + 1; i++) {

if(lumaLevel > SpsLadfIntervalLowerBound[i + 1])

qpOffset = sps_ladf_qp_offset[i] (8-1071)

si no

break}

- De lo contrario, qpOffset se establece igual a 0.

Las variables Q_{pQ} y Q_{pP} se establecen iguales a los valores Q_{pY} de las unidades de codificación que incluyen los bloques de codificación que contienen la muestra $q_{0,0}$ y $p_{0,0}$, respectivamente. En este caso, Q_{pQ} y Q_{pP} representan los valores de QP de luma respectivamente, para distinguir dos valores de QP de luma, se representarán mediante Q_{pYQ} y Q_{pYP} respectivamente en la parte reivindicación//

La variable qP se deriva del siguiente modo:

$$qP = ((Q_{pQ} + Q_{pP} + 1) \gg 1) + qpOffset \quad (8-1072)$$

El valor de la variable β' se determina como se especifica en la Tabla 8-18 basado en el parámetro de cuantificación Q derivado del siguiente modo:

$$Q = \text{Clip3}(0, 63, qP + (\text{slice_beta_offset_div2} \ll 1)) \quad (8-1073)$$

donde `slice_beta_offset_div2` es el valor del elemento de sintaxis `slice_beta_offset_div2` para el segmento que contiene la muestra $q_{0,0}$.

La variable β se deriva del siguiente modo:

$$\beta = \beta' * (1 \ll (\text{BitDepth}_Y - 8)) \quad (8-1074)$$

El valor de la variable tc' se determina tal como se especifica en la Tabla 8-18 basado en el parámetro de cuantificación Q derivado del siguiente modo:

$$Q = \text{Clip3}(0, 65, qP + 2 * (bS - 1) + (\text{slice_tc_offset_div2} \ll 1)) \quad (8-1075)$$

donde `slice_tc_offset_div2` es el valor del elemento de sintaxis `slice_tc_offset_div2` para el segmento que contiene la muestra $q_{0,0}$.

La variable tc se deriva del siguiente modo:

$$tc = \text{BitDepth}_Y < 10 ? (tc' + 2) \gg (10 - \text{BitDepth}_Y) : tc' * (1 \ll (\text{BitDepth}_Y - 10)) \quad (8-1076)$$

Se aplican las siguientes etapas ordenadas:

1. Las variables dp_0 , dp_3 , dq_0 y dq_3 se derivan del siguiente modo:

$$dp_0 = \text{Abs}(p_{2,0} - 2 * p_{1,0} + p_{0,0}) \quad (8-1077)$$

$$dp_3 = \text{Abs}(p_{2,3} - 2 * p_{1,3} + p_{0,3}) \quad (8-1078)$$

$$dq_0 = \text{Abs}(q_{2,0} - 2 * q_{1,0} + q_{0,0}) \quad (8-1079)$$

$$dq_3 = \text{Abs}(q_{2,3} - 2 * q_{1,3} + q_{0,3}) \quad (8-1080)$$

2. Cuando `maxFilterLengthP` y `maxFilterLengthQ` son iguales o mayores que 3, las variables sp_0 , sq_0 , spq_0 , sp_3 , sq_3 y spq_3 se derivan del siguiente modo:

$$sp_0 = \text{Abs}(p_{3,0} - p_{0,0}) \quad (8-1081)$$

$$sq_0 = \text{Abs}(q_{0,0} - q_{3,0}) \quad (8-1082)$$

$$spq_0 = \text{Abs}(p_{0,0} - q_{0,0}) \quad (8-1083)$$

$$sp_3 = \text{Abs}(p_{3,3} - p_{0,3}) \quad (8-1084)$$

$$sq_3 = \text{Abs}(q_{0,3} - q_{3,3}) \quad (8-1085)$$

$$spq_3 = \text{Abs}(p_{0,3} - q_{0,3}) \quad (8-1086)$$

3. Las variables `sidePisLargeBlk` y `sideQisLargeBlk` se establecen iguales a 0.

4. Cuando `maxFilterLengthP` es mayor que 3, `sidePisLargeBlk` se establece igual a 1:

5. Cuando `maxFilterLengthQ` es mayor que 3, `sideQisLargeBlk` se establece igual a 1:

6. Cuando `edgeType` es igual a `EDGE_HOR` y $(yCb + yBl) \% CtbSizeY$ es igual a 0, `sidePisLargeBlk` se establece igual a 0.

7. Las variables `dSam0` y `dSam3` se inicializan en 0.

8. Cuando `sidePisLargeBlk` o `sideQisLargeBlk` es mayor que 0, se aplica lo siguiente:

a. se derivan las variables `dp0L`, `dp3L` y `maxFilterLengthP` se modifica del siguiente modo:

– si sidePisLargeBlk es igual a 1, se aplica lo siguiente:

$$5 \quad dp0L = (dp0 + Abs(p_{5,0} - 2 * p_{4,0} + P_{3,0}) + 1) \gg 1 \quad (8-1087)$$

$$dp3L = (dp3 + Abs(p_{5,3} - 2 * p_{4,3} + P_{3,3}) + 1) \gg 1 \quad (8-1088)$$

– De lo contrario, se aplica lo siguiente:

$$10 \quad dp0L = dp0 \quad (8-1089)$$

$$dp3L = dp3 \quad (8-1090)$$

$$15 \quad maxFilterLengthP = 3 \quad (8-1091)$$

b. las variables dp0L y dp3L se derivan del siguiente modo:

– si sideQisLargeBlk es igual a 1, se aplica lo siguiente:

$$20 \quad dq0L = (dq0 + Abs(q_{5,0} - 2 * q_{4,0} + q_{3,0}) + 1) \gg 1 \quad (8-1092)$$

$$dq3L = (dq3 + Abs(q_{5,3} - 2 * q_{4,3} + q_{3,3}) + 1) \gg 1 \quad (8-1093)$$

– De lo contrario, se aplica lo siguiente:

$$25 \quad dq0L = dq0 \quad (8-1094)$$

$$dq3L = dq3 \quad (8-1095)$$

30 c. las variables dpq0L, dpq3L, y dL se derivan del siguiente modo:

$$dpq0L = dp0L + dq0L \quad (8-1096)$$

$$35 \quad dpq3L = dp3L + dq3L \quad (8-1097)$$

$$dL = dpq0L + dpq3L \quad (8-1098)$$

d. cuando dL es menor que β , se aplican las siguientes etapas ordenadas:

40 i. la variable dpq se establece igual a $2 * dpq0L$.

ii. La variable sp se establece igual a sp0, la variable sq se establece igual a sq0 y la variable spq se establece igual a spq0.

45 iii. Las variables p0 p3 q0 y q3 se inicializan primero en 0 y, a continuación, se modifican según sidePisLargeBlk y sideQisLargeBlk del siguiente modo:

– Cuando sidePisLargeBlk es igual a 1, se aplica lo siguiente:

$$50 \quad p3 = p_{3,0} \quad (8-1099)$$

$$p0 = p_{maxFilterLengthP,0} \quad (8-1100)$$

– Cuando sideQisLargeBlk es igual a 1, se aplica lo siguiente:

$$55 \quad q3 = q_{3,0} \quad (8-1101)$$

$$q0 = q_{maxFilterLengthQ,0} \quad (8-1102)$$

60 iv. Para la ubicación de la muestra (xCb + xBl, yCb + yBl), se invoca al proceso de decisión para una muestra de luma, tal como se especifica en la cláusula 8.8.3.6.5, con los valores de muestra p0, p3, q0, q3, las variables dpq, sp, sq, spq, sidePisLargeBlk, sideQisLargeBlk, β y tc como entradas, y la salida se asigna a la decisión dSam0.

65 v. La variable dpq se establece igual a $2 * dpq3L$.

vi. La variable sp se establece igual a sp_3 , la variable sq se establece igual a sq_3 y la variable spq se establece igual a spq_3 .

vii. Las variables p_0 , p_3 , q_0 y q_3 se inicializan primero en 0 y, a continuación, se modifican según $sidePisLargeBlk$ y $sideQisLargeBlk$ del siguiente modo:

– Cuando $sidePisLargeBlk$ es igual a 1, se aplica lo siguiente:

$$p_3 = p_{3,3} \quad (8-1103)$$

$$p_0 = p_{\maxFilterLengthP,3} \quad (8-1104)$$

– Cuando $sideQisLargeBlk$ es igual a 1, se aplica lo siguiente:

$$q_3 = q_{3,3} \quad (8-1105)$$

$$q_0 = q_{\maxFilterLengthQ,3} \quad (8-1106)$$

viii. Cuando $edgeType$ es igual a $EDGE_VER$ para la ubicación de la muestra ($x_{Cb} + x_{Bl}$, $y_{Cb} + y_{Bl} + 3$) o cuando $edgeType$ es igual a $EDGE_HOR$ para la ubicación de la muestra ($x_{Cb} + x_{Bl} + 3$, $y_{Cb} + y_{Bl}$), el proceso de decisión para la muestra de luma, tal como se especifica en la cláusula 8.8.3.6.5, se invoca con los valores de la muestra p_0 , p_3 , q_0 , q_3 , la variables dpq , sp , sq , spq , $sidePisLargeBlk$, $sideQisLargeBlk$, β y tc como entradas, y la salida se asigna a la decisión $dSam3$.

9. Las variables dE , dEp y dEq se derivan del siguiente modo:

– si $dSam0$ y $dSam3$ son ambos iguales a 1, la variable dE se establece igual a 3, dEp se establece igual a 1 y dEq se establece igual a 1.

–De lo contrario, se aplican los siguientes etapas ordenadas:

a. las variables $dpq0$, $dpq3$, dp , dq y d se derivan del siguiente modo:

$$dpq0 = dp0 + dq0 \quad (8-1107)$$

$$dpq3 = dp3 + dq3 \quad (8-1108)$$

$$dp = dp0 + dp3 \quad (8-1109)$$

$$dq = dq0 + dq3 \quad (8-1110)$$

$$d = dpq0 + dpq3 \quad (8-1111)$$

b. las variables dE , dEp , dEq , $sidePisLargeBlk$ y $sideQisLargeBlk$ se establecen iguales a 0.

c. Cuando d es menor que β y tanto \maxFilterLengthP como \maxFilterLengthQ son mayores que 2, se aplican las siguientes etapas ordenadas:

i. la variable dpq se establece igual a $2 * dpq0$.

ii. La variable sp se establece igual a $sp0$, la variable sq se establece igual a $sq0$ y la variable spq se establece igual a $spq0$.

iii. Para la ubicación de la muestra ($x_{Cb} + x_{Bl}$, $y_{Cb} + y_{Bl}$), se invoca al proceso de decisión para una muestra de luma, tal como se especifica en la cláusula 8.8.3.6.5, con las variables p_0 , p_3 , q_0 , q_3 todas establecidas igual a 0, las variables dpq , sp , sq , spq , $sidePisLargeBlk$, $sideQisLargeBlk$, β y tc como entradas, y se asigna la salida a la decisión $dSam0$.

iv. La variable dpq se establece igual a $2 * dpq3$.

v. La variable sp se establece igual a sp_3 , la variable sq se establece igual a sq_3 y la variable spq se establece igual a spq_3 .

vi. Cuando $edgeType$ es igual a $EDGE_VER$ para la ubicación de la muestra ($x_{Cb} + x_{Bl}$, $y_{Cb} + y_{Bl} + 3$) o cuando $edgeType$ es igual a $EDGE_HOR$ para la ubicación de la muestra ($x_{Cb} + x_{Bl} + 3$, $y_{Cb} + y_{Bl}$), el proceso de decisión de una muestra, tal como se especifica en la cláusula 8.8.3.6.5, se invoca con las variables p_0 , p_3 , q_0 , q_3 todas establecidas igual a 0, las variables dpq , sp , sq , spq , $sidePisLargeBlk$, $sideQisLargeBlk$, β y tc como entradas, y la salida se asigna a la decisión $dSam3$.

d. Cuando d es menor que β , se aplican las siguientes etapas ordenadas:

i. la variable dE se establece igual a 1.

ii. cuando $dSam0$ es igual a 1 y $dSam3$ es igual a 1, la variable dE se establece igual a 2.

iii. Cuando $maxFilterLengthP$ es mayor que 1, $maxFilterLengthQ$ es mayor que 1 y dp es menor que $(\beta + (\beta \gg 1)) \gg 3$, la variable dEp se establece igual a 1.

iv. Cuando $maxFilterLengthP$ es mayor que 1, $maxFilterLengthQ$ es mayor que 1 y dq es menor que $(\beta + (\beta \gg 1)) \gg 3$, la variable dEq se establece igual a 1.

Tabla 8-18: Derivación de las variables de umbral β' y tc' a partir de la entrada Q

Q	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
β'	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6
tc'	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Q	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
β'	7	8	9	10	11	12	13	14	15	16	17	18	20	22	24	26	28
tc'	0	3	4	4	4	4	5	5	5	5	7	7	8	9	10	10	11
Q	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
β'	30	32	34	36	38	40	42	44	46	48	50	52	54	56	58	60	62
tc'	13	14	15	17	19	21	24	25	29	33	36	41	45	51	57	64	71
Q	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65		
β'	64	66	68	70	72	74	76	78	80	82	84	86	88	-	-		
tc'	80	89	100	112	125	141	157	177	198	222	250	280	314	352	395		

8.8.3.6.2 Proceso de filtrado para bordes de bloques de luma

Las entradas de este proceso son:

- una matriz de muestra de imagen $recPicture$,
- una ubicación (xCb, yCb) que especifica la muestra superior izquierda del bloque de codificación actual en relación con la muestra superior izquierda de la imagen actual,
- una ubicación (xBl, yBl) que especifica la muestra superior izquierda del bloque actual en relación con la muestra superior izquierda del bloque de codificación actual,
- una variable $edgeType$ que especifica si se filtra un borde vertical ($EDGE_VER$) u horizontal ($EDGE_HOR$),
- las variables dE , dEp y dEq que contienen decisiones,
- las variables $maxFilterLengthP$ y $maxFilterLengthQ$ que contienen las longitudes máximas de filtro,
- la variable tc .

La salida de este proceso es la matriz de muestra de imagen modificada $recPicture$.

Dependiendo del valor de $edgeType$, se aplica lo siguiente:

- Si $edgeType$ es igual a $EDGE_VER$, se aplican las siguientes etapas ordenadas:

1. Los valores de muestra $p_{i,k}$ y $q_{i,k}$ con $i = 0.. maxFilterLengthP$, $j = 0.. maxFilterLengthQ$ y $k = 0..3$ se derivan del siguiente modo:

$$q_{i,k} = recPicture_{[xCb + xB1 + j][yCb + yB1 + k]} \quad (8-1112)$$

$$p_{i,k} = \text{recPicture}_{\lfloor xCb + xB1 - i - 1 \rfloor} \lfloor yCb + yB1 + k \rfloor \quad (8-1113)$$

2. Cuando dE no es igual a 0 y dE no es igual a 3, para cada ubicación de muestra (xCb + xBl, yCb + yBl + k), k = 0..3, se aplican las siguientes etapas ordenadas:

a. El proceso de filtrado para una muestra de luma usando filtros cortos, tal como se especifica en la cláusula 8.8.3.6.6, se invoca con los valores de muestra $p_{i,k}$, $q_{i,k}$ con $i = 0..3$, las ubicaciones (xP_i, yP_i) establecida igual a (xCb + xBl - i - 1, yCb + yBl + k) y (xQ_i, yQ_i) establecida igual a (xCb + xBl + i, yCb + yBl + k) con $i = 0..2$, la decisión dE, las variables dEp y dEq y la variable tc como entradas, y el número de muestras filtradas nDp y nDq de cada lado del límite del bloque y los valores de muestra filtrada p_i' y q_i' como salidas.

b. Cuando nDp es mayor que 0, los valores de muestra filtrada p_i' con $i = 0.. nDp - 1$ sustituyen las muestras correspondientes dentro de la matriz de muestra recPicture del siguiente modo:

$$\text{recPicture}_{\lfloor xCb + xB1 - i - 1 \rfloor} \lfloor yCb + yB1 + k \rfloor = p_i' \quad (8-1114)$$

c. Cuando nDq es mayor que 0, los valores de muestra filtrada q_j' con $j = 0..nDq - 1$ sustituyen las muestras correspondientes dentro de la matriz de muestras recPicture del siguiente modo:

$$\text{recPicture}_{\lfloor xCb + xB1 + j \rfloor} \lfloor yCb + yB1 + k \rfloor = q_j' \quad (8-1115)$$

3. Cuando dE es igual a 3, para cada ubicación de muestra (xCb + xBl, yCb + yBl + k), k = 0..3, se aplican las siguientes etapas ordenadas:

a. El proceso de filtrado de una muestra de luma que utiliza filtros largos, tal como se especifica en la cláusula 8.8.3.6.7, se invoca con los valores de muestra $p_{i,k}$, $q_{j,k}$ con $i = 0..maxFilterLengthP$ y $j = 0..maxFilterLengthQ$, las ubicaciones (xP_i, yP_i) establecida como igual a (xCb + xBl - i - 1, yCb + yBl + k) con $i = 0..maxFilterLengthP - 1$ y (xQ_j, yQ_j) establecida como igual a (xCb + xBl + j, yCb + yBl + k) con $j = 0..maxFilterLengthQ - 1$, las variables maxFilterLengthP, maxFilterLengthQ y tc como entradas y los valores p_i' y q_j' de las muestras filtradas como salidas.

b. Los valores de muestra filtrada p_i' con $i = 0.. maxFilterLengthP - 1$ sustituyen las muestras correspondientes dentro de la matriz de muestras recPicture del siguiente modo:

$$\text{recPicture}_{\lfloor xCb + xB1 - i - 1 \rfloor} \lfloor yCb + yB1 + k \rfloor = p_i' \quad (8-1116)$$

c. Los valores de la muestra filtrada q_j' con $j = 0.. maxFilterLengthQ - 1$ sustituyen las muestras correspondientes dentro de la matriz de muestras recPicture del siguiente modo:

$$\text{recPicture}_{\lfloor xCb + xB1 + j \rfloor} \lfloor yCb + yB1 + k \rfloor = q_j' \quad (8-1117)$$

– De lo contrario (edgeType es igual a EDGE_HOR), se aplican las siguientes etapas ordenadas:

1. Los valores de muestra $p_{i,k}$ y $q_{j,k}$ con $i = 0.. maxFilterLengthP$, $j = 0.. maxFilterLengthQ$ y $k = 0..3$ se derivan del siguiente modo:

$$q_{j,k} = \text{recPicture}_{\lfloor xCb + xB1 + k \rfloor} \lfloor yCb + yB1 + j \rfloor \quad (8-1118)$$

$$p_{i,k} = \text{recPicture}_{\lfloor xCb + xB1 + k \rfloor} \lfloor yCb + yB1 - i - 1 \rfloor \quad (8-1119)$$

2. Cuando dE no es igual a 0 y dE no es igual a 3, para cada ubicación de muestra (xCb + xBl + k, yCb + yBl), k = 0..3, se aplican las siguientes etapas ordenadas:

a. El proceso de filtrado de una muestra de luma usando filtros cortos, tal como se especifica en la cláusula 8.8.3.6.6, se invoca con los valores de muestra $p_{i,k}$, $q_{i,k}$ con $i = 0..3$, las ubicaciones (xP_i, yP_i) establecida igual a (xCb + xBl + k, yCb + yBl - i - 1) y (xQ_i, yQ_i) establecida igual a (xCb + xBl + k, yCb + yBl + i) con $i = 0..2$, la decisión dE, las variables dEp y dEq, y la variable tc como entradas, y el número de muestras filtradas nDp y nDq de cada lado del límite de bloque y los valores de la muestra filtrada p_i' y q_j' como salidas.

b. Cuando nDp es mayor que 0, los valores de muestra filtrada p_i' con $i = 0.. nDp - 1$ sustituyen las muestras correspondientes dentro de la matriz de muestra recPicture del siguiente modo:

$$\text{recPicture}_{\lfloor xCb + xB1 + k \rfloor} \lfloor yCb + yB1 - i - 1 \rfloor = p_i' \quad (8-1120)$$

c. Cuando nDq es mayor que 0, los valores de muestra filtrada q_j' con $j = 0..nDq - 1$ sustituyen las muestras correspondientes dentro de la matriz de muestra recPicture del siguiente modo:

$$\text{recPicture}[\text{xCb} + \text{xBl} + \text{k}] [\text{yCb} + \text{yBl} + \text{j}] = \text{q}'_j \quad (8-1121)$$

3. Cuando dE es igual a 3, para cada ubicación de muestra $(\text{xCb} + \text{xBl} + \text{k}, \text{yCb} + \text{yBl})$, $\text{k} = 0..3$, se aplican las siguientes etapas ordenadas:

5 a. El proceso de filtrado para una muestra de luma usando filtros largos, tal como se especifica en la cláusula 8.8.3.6.7, se invoca con los valores de muestra $p_{i,k}$, $q_{i,k}$ con $i = 0.. \text{maxFilterLengthP}$ y $j = 0.. \text{maxFilterLengthQ}$, las ubicaciones $(\text{xP}_i, \text{yP}_i)$ establecida igual a $(\text{xCb} + \text{xBl} + \text{k}, \text{yCb} + \text{yBl} - i - 1)$ con $i = 0.. \text{maxFilterLengthP} - 1$ y $(\text{xQ}_j, \text{yQ}_j)$ establecida igual a $(\text{xCb} + \text{xBl} + \text{k}, \text{yCb} + \text{yBl} + \text{j})$ con $j = 0.. \text{maxFilterLengthQ} - 1$, las variables maxFilterLengthP , maxFilterLengthQ , y la variable tc como entradas, y los valores de muestra filtrada p'_i y q'_j como salidas.

b. Los valores de muestra filtrada p'_i con $i = 0.. \text{maxFilterLengthP} - 1$ sustituyen las muestras correspondientes dentro de la matriz de muestras recPicture del siguiente modo:

$$15 \quad \text{recPicture}[\text{xCb} + \text{xBl} + \text{k}] [\text{yCb} + \text{yBl} - i - 1] = p'_i \quad (8-1122)$$

c. Los valores de la muestra filtrada q'_j con $j = 0.. \text{maxFilterLengthQ} - 1$ sustituyen las muestras correspondientes dentro de la matriz de muestras recPicture del siguiente modo:

$$20 \quad \text{recPicture}[\text{xCb} + \text{xBl} + \text{k}] [\text{yCb} + \text{yBl} + \text{j}] = q'_j \quad (8-1123)$$

8.8.3.6.3 Proceso de decisión para bordes de bloques de croma

Este proceso solo se invoca cuando ChromaArrayType no es igual a 0.

25 Las entradas de este proceso son:

- una matriz de muestra de imagen de croma recPicture ,
- 30 – una ubicación de croma (xCb, yCb) que especifica la muestra superior izquierda del bloque de codificación de croma actual en relación con la muestra de croma superior izquierda de la imagen actual,
- una ubicación de croma (xBl, yBl) que especifica la muestra superior izquierda del bloque de croma actual en relación con la muestra superior izquierda del bloque de codificación de croma actual,
- 35 – una variable edgeType que especifica si se filtra un borde vertical (EDGE_VER) u horizontal (EDGE_HOR),
- una variable cIdx que especifica el índice del componente de color,
- 40 – una variable cQpPicOffset que especifica el desplazamiento del parámetro de cuantificación de croma a nivel de imagen,
- una variable bS que especifica la intensidad de filtrado de límites,
- 45 – una variable $\text{maxFilterLengthCbCr}$.

Las salidas de este proceso son

- 50 – la variable modificada $\text{maxFilterLengthCbCr}$,
- la variable tc .

La variable maxK se deriva del siguiente modo:

- 55 – Si edgeType es igual a EDGE_VER , se aplica lo siguiente:

$$\text{maxK} = (\text{SubHeightC} == 1) ? 3 : 1 \quad (8-1124)$$

- 60 – De lo contrario (edgeType es igual a EDGE_HOR), se aplica lo siguiente:

$$\text{maxK} = (\text{SubWidthC} == 1) ? 3 : 1 \quad (8-1125)$$

Los valores p_i y q_j con $i = 0.. \text{maxFilterLengthCbCr}$ y $k = 0.. \text{maxK}$ se derivan del siguiente modo:

- 65 – Si edgeType es igual a EDGE_VER , se aplica lo siguiente:

$$q_{i,k} = \text{recPicture} [x_{CB} + x_{B1} + i] [y_{Cb} + y_{B1} + k] \quad (8-1126)$$

$$p_{i,k} = \text{recPicture} [x_{CB} + x_{B1} - i - 1] [y_{Cb} + y_{B1} + k] \quad (8-1127)$$

5

$$\text{subSampleC} = \text{SubHeightC} \quad (8-1128)$$

– De lo contrario (edgeType es igual a EDGE_HOR), se aplica lo siguiente:

10

$$q_{i,k} = \text{recPicture} [x_{Cb} + x_{B1} + k] [y_{Cb} + y_{B1} + i] \quad (8-1129)$$

$$p_{i,k} = \text{recPicture} [x_{Cb} + x_{B1} + k] [y_{Cb} + y_{B1} - i - 1] \quad (8-1130)$$

15

$$\text{subSampleC} = \text{SubWidthC} \quad (8-1131)$$

Las variables Q_{pQ} y Q_{pP} se establecen iguales a los valores Q_{pY} de las unidades de codificación que incluyen los bloques de codificación que contienen la muestra $q_{0,0}$ y $p_{0,0}$, respectivamente.

La variable Q_{pc} se deriva del siguiente modo:

20

$$q_{Pi} = \text{Clip3}(0, 63, ((Q_{pQ} + Q_{pP} + 1) \gg 1) + c_{QpPicOffset}) \quad (8-1132)$$

$$Q_{pc} = \text{ChromaQpTable}[c_{ldx} - 1] [q_{Pi}] \quad (8-1133)$$

25

NOTA: la variable $c_{QpPicOffset}$ proporciona un ajuste para el valor de $pps_cb_qp_offset$ o $pps_cr_qp_offset$, según si el componente de croma filtrado es el componente de Cb o Cr. Sin embargo, para evitar la necesidad de variar la cantidad del ajuste dentro de la imagen, el proceso de filtrado no incluye un ajuste para el valor de $slice_cb_qp_offset$ o $slice_cr_qp_offset$ ni (cuando $cu_chroma_qp_offset_enabled_flag$ es igual a 1) para el valor de $CuQpOffset_{Cb}$, $CuQpOffset_{Cr}$, o $CuQpOffset_{CbCr}$.

30

El valor de la variable β' se determina como se especifica en la Tabla 8-18 basado en el parámetro de cuantificación Q derivado del siguiente modo:

$$Q = \text{Clip3}(0, 63, Q_{pc} + (\text{slice_beta_offset_div2} \ll 1)) \quad (8-1134)$$

35

donde $slice_beta_offset_div2$ es el valor del elemento de sintaxis $slice_beta_offset_div2$ para el segmento que contiene la muestra $q_{0,0}$.

La variable β se deriva del siguiente modo:

40

$$\beta = \beta' * (1 \ll (\text{BitDepth}_C - 8)) \quad (8-1135)$$

El valor de la variable tc' se determina tal como se especifica en la Tabla 8-18 basado en el parámetro de cuantificación Q de croma derivado del siguiente modo:

45

$$Q = \text{Clip3}(0, 65, Q_{pc} + 2 * (bS - 1) + (\text{slice_tc_offset_div2} \ll 1)) \quad (8-1136)$$

donde $slice_tc_offset_div2$ es el valor del elemento de sintaxis $slice_tc_offset_div2$ para el segmento que contiene la muestra $q_{0,0}$.

50

La variable tc se deriva del siguiente modo:

$$tc = (\text{BitDepth}_C < 10) ? (tc' + 2) \gg (10 - \text{BitDepth}_C) : tc' * (1 \ll (\text{BitDepth}_C - 8)) \quad (8-1137)$$

55

Cuando $\text{maxFilterLengthCbCr}$ es igual a 1 y bS no es igual a 2, $\text{maxFilterLengthCbCr}$ se establece igual a 0.

Cuando $\text{maxFilterLengthCbCr}$ es igual a 3, se aplican las siguientes etapas ordenadas:

1. Las variables $n1$, $dpq0$, $dpq1$, dp , dq y d se derivan del siguiente modo:

60

$$n1 = (\text{subSampleC} = 2) ? 1 : 3 \quad (8-1138)$$

$$dp0 = \text{Abs}(p_{2,0} - 2 * p_{1,0} + p_{0,0}) \quad (8-1139)$$

$$dp1 = \text{Abs}(p_{2,n1} - 2 * p_{1,n1} + p_{0,n1}) \quad (8-1140)$$

65

$$dq0 = \text{Abs}(q_{2,0} - 2 * q_{1,0} + q_{0,0}) \quad (8-1141)$$

$$dq1 = \text{Abs}(q_{2,n1} - 2 * q_{1,n1} + q_{0,n1}) \quad (8-1142)$$

5

$$dpq0 = dp0 + dq0 \quad (8-1143)$$

$$dpq1 = dp1 + dq1 \quad (8-1144)$$

$$dp = dp0 + dp1 \quad (8-1145)$$

10

$$dq = dq0 + dq1 \quad (8-1146)$$

$$d = dpq0 + dpq1 \quad (8-1147)$$

15

2. Las variables dSam0 y dSam1 se establecen iguales a 0.

3. Cuando d es menor que β , se aplican las siguientes etapas ordenadas:

20

a. La variable dpq se establece igual a $2 * dpq0$.

b. La variable dSam0 se deriva invocando al proceso de decisión para una muestra de croma, tal como se especifica en la cláusula 8.8.3.6.8, para la ubicación de la muestra ($x_{Cb} + x_{Bl}$, $y_{Cb} + y_{Bl}$) con valores de la muestra $p_{0,0}$, $P_{3,0}$, $q_{0,0}$ y $q_{3,0}$, las variables dpq, β y t_c como entradas, y la salida se asigna a la decisión dSam0.

25

c. La variable dpq se establece igual a $2 * dpq1$.

d. La variable dSam1 se modifica del siguiente modo:

30

- Si edgeType es igual a EDGE_VER, para la ubicación de la muestra ($x_{Cb} + x_{Bl}$, $y_{Cb} + y_{Bl} + n1$), se invoca el proceso de decisión para una muestra de croma, tal como se especifica en la cláusula 8.8.3.6.8, con los valores de muestra $p_{0,n1}$, $p_{3,n1}$, $q_{0,n1}$ y $q_{3,n1}$, las variables dpq, β y t_c como entradas, y la salida se asigna a la decisión dSam1.

35

- De lo contrario (edgeType es igual a EDGE_HOR), para la ubicación de la muestra ($x_{Cb} + x_{Bl} + n1$, $y_{Cb} + y_{Bl}$), se invoca el proceso de decisión para una muestra de croma tal como se especifica en la cláusula 8.8.3.6.8 con los valores de muestra $p_{0,n1}$, $p_{3,n1}$, $q_{0,n1}$ y $q_{3,n1}$, las variables dpq, β y t_c como entradas, y la salida se asigna a la decisión dSam1.

4. La variable maxFilterLengthCbCr se modifica del siguiente modo:

40

- Si dSam0 es igual a 1 y dSam1 es igual a 1, maxFilterLengthCbCr se establece igual a 3.

- De lo contrario, maxFilterLengthCbCr se establece igual a 1.

45

Cabe señalar que, tal como se ilustra en 8-1132 y 8-1133 de la sección 8.8.3.6.3, en el enfoque convencional, un QP de luma (p. ej., q_{Pi}) se deriva basado en un promedio de los QP de luma de dos bloques adyacentes (p. ej., Q_{p0} y Q_{p1}); a continuación, un QP de croma (p. ej., Q_{pc}) se deriva del QP de luma (p. ej., q_{Pi}) (usando una LUT). La realización de la presente descripción mejora con respecto al enfoque convencional y los detalles sobre cómo se lleva a cabo un proceso de decisión para el borde del bloque de croma se describirán a continuación.

8.8.3.6.4 Proceso de filtrado para bordes de bloques de croma

50

Este proceso solo se invoca cuando ChromaArrayType no es igual a 0.

Las entradas de este proceso son:

55

- una matriz de muestra de imagen de croma recPicture,

- una ubicación de croma (x_{Cb} , y_{Cb}) que especifica la muestra superior izquierda del bloque de codificación de croma actual en relación con la muestra de croma superior izquierda de la imagen actual,

60

- una ubicación de croma (x_{Bl} , y_{Bl}) que especifica la muestra superior izquierda del bloque de croma actual en relación con la muestra superior izquierda del bloque de codificación de croma actual,

- una variable edgeType que especifica si se filtra un borde vertical (EDGE_VER) u horizontal (EDGE_HOR),

65

- una variable maxFilterLengthCbCr que contiene la longitud máxima de filtro de croma,

– la variable tc.

La salida de este proceso es la matriz de muestra de imagen de croma modificada recPicture.

5 La variable maxK se deriva del siguiente modo:

– Si edgeType es igual a EDGE_VER, se aplica lo siguiente:

$$\text{maxK} = (\text{SubHeightC} = 1) ? 3 : 1 \quad (8-1148)$$

10

– De lo contrario (edgeType es igual a EDGE_HOR), se aplica lo siguiente:

$$\text{maxK} = (\text{SubWidthC} = 1) ? 3 : 1 \quad (8-1149)$$

15

Los valores p_i y q_i con $i = 0..maxFilterLengthCbCr$ y $k = 0..maxK$ se derivan del siguiente modo:

– Si edgeType es igual a EDGE_VER, se aplica lo siguiente:

20

$$q_{i,k} = \text{recPicture} [\text{xCb} + \text{xBl} + i] [\text{yCb} + \text{yBl} + k] \quad (8-1150)$$

$$p_{i,k} = \text{recPicture} [\text{xCb} + \text{xBl} - i - 1] [\text{yCb} + \text{yBl} + k] \quad (8-1151)$$

– De lo contrario (edgeType es igual a EDGE_HOR), se aplica lo siguiente:

25

$$q_{i,k} = \text{recPicture} [\text{xCb} + \text{xBl} + k] [\text{yCb} + \text{yBl} + i] \quad (8-1152)$$

$$p_{i,k} = \text{recPicture} [\text{xCb} + \text{xBl} + k] [\text{yCb} + \text{yBl} - i - 1] \quad (8-1153)$$

30

Dependiendo del valor de edgeType, se aplica lo siguiente:

– Si edgeType es igual a EDGE_VER, para cada ubicación de muestra ($\text{xCb} + \text{xBl}$, $\text{yCb} + \text{yBl} + k$), $k = 0..maxK$, se aplican las siguientes etapas ordenadas:

35

1. El proceso de filtrado para una muestra de croma, tal como se especifica en la cláusula 8.8.3.6.9, se invoca con la variable maxFilterLengthCbCr, los valores de muestra $p_{i,k}$, $q_{i,k}$ con $i = 0..maxFilterLengthCbCr$, las ubicaciones ($\text{xCb} + \text{xBl} - i - 1$, $\text{yCb} + \text{yBl} + k$) y ($\text{xCb} + \text{xBl} + i$, $\text{yCb} + \text{yBl} + k$) con $i = 0..maxFilterLengthCbCr - 1$, y la variable tc como entradas y los valores de muestra filtrada p'_i y q'_i con $i = 0..maxFilterLengthCbCr - 1$ como salidas.

40

2. Los valores de muestra filtrada p'_i y q'_i con $i = 0..maxFilterLengthCbCr - 1$ sustituyen las muestras correspondientes dentro de la matriz de muestras recPicture del siguiente modo:

$$\text{recPicture} [\text{xCb} + \text{xBl} + i] [\text{yCb} + \text{yBl} + k] = q'_i \quad (8-1154)$$

45

$$\text{recPicture} [\text{xCb} + \text{xBl} - i - 1] [\text{yCb} + \text{yBl} + k] = p'_i \quad (8-1155)$$

– De lo contrario (edgeType es igual a EDGE_HOR), para cada ubicación de muestra ($\text{xCb} + \text{xBl} + k$, $\text{yCb} + \text{yBl}$), $k = 0..maxK$, se aplican las siguientes etapas ordenadas:

50

1. El proceso de filtrado para una muestra de croma, tal como se especifica en la cláusula 8.8.3.6.9, se invoca con la variable maxFilterLengthCbCr, los valores de muestra $p_{i,k}$, $q_{i,k}$, con $i = 0..maxFilterLengthCbCr$, las ubicaciones ($\text{xCb} + \text{xBl} + k$, $\text{yCb} + \text{yBl} - i - 1$) y ($\text{xCb} + \text{xBl} + k$, $\text{yCb} + \text{yBl} + i$), y la variable tc como entradas, y los valores de muestra filtrada p'_i y q'_i como salidas.

55

2. Los valores de muestra filtrada p'_i y q'_i sustituyen las muestras correspondientes dentro de la matriz de muestra recPicture del siguiente modo:

$$\text{recPicture} [\text{xCb} + \text{xBl} + k] [\text{yCb} + \text{yBl} + i] = q'_i \quad (8-1156)$$

60

$$\text{recPicture} [\text{xCb} + \text{xBl} + k] [\text{yCb} + \text{yBl} - i - 1] = p'_i \quad (8-1157)$$

Cabe señalar que el proceso de filtrado para el borde de bloque de croma basado en el parámetro de umbral (p. ej., la variable tc) se puede encontrar en documentos tales como en la sección 8.8.3.6.4 anterior de la especificación de la VVC, y no se repetirá en lo que sigue.

65

8.8.3.6.5 Proceso de decisión para una muestra de luma

Las entradas de este proceso son:

- 5 - los valores de muestra p_0 , p_3 , q_0 y q_3 ,
- las variables dpq , sp , sq , spq , $sidePisLargeBlk$, $sideQisLargeBlk$, β y tc .

La salida de este proceso es la variable $dSam$ que contiene una decisión.

10 Las variables sp y sq se modifican del siguiente modo:

- Cuando $sidePisLargeBlk$ es igual a 1, se aplica lo siguiente:

$$15 \quad sp = (sp + Abs(p_3 - p_0) + 1) \gg 1 \quad (8-1158)$$

- Cuando $sideQisLargeBlk$ es igual a 1, se aplica lo siguiente:

$$sq = (sq + Abs(p_3 - p_0) + 1) \gg 1 \quad (8-1158)$$

20 La variable $sThr$ se deriva del siguiente modo:

- Si $sidePisLargeBlk$ es igual a 1 o $SideQisLargeBlk$ es igual a 1, se aplica lo siguiente:

$$25 \quad sThr = 3 * \beta \gg 5 \quad (8-1160)$$

- De lo contrario, se aplica lo siguiente:

$$sThr = \beta \gg 3 \quad (8-1161)$$

30 La variable $dSam$ se especifica del siguiente modo:

-Si se cumplen todas las condiciones siguientes, $dSam$ se establece igual a 1:

- 35 - dpq es menor que $(\beta \gg 2)$,
- $sp + sq$ es menor que $sThr$,
- spq es menor que $(5 * tc + 1) \gg 1$.

- 40 - De lo contrario, $dSam$ se establece igual a 0.

8.8.3.6.6 Proceso de filtrado para una muestra de luma usando filtros cortos

45 Las entradas de este proceso son:

- los valores de muestra p_i y q_i con $i = 0..3$,
- las ubicaciones de p_i y q_i , (xP_i, yP_i) y (xQ_i, yQ_i) con $i = 0..2$,
- 50 - una variable dE ,
- las variables dEp y dEq que contienen decisiones para filtrar las muestras p_1 y q_1 , respectivamente,
- 55 - una variable tc .

Las salidas de este proceso son:

- el número de muestras filtradas nDp y nDq ,
- 60 - los valores de muestra filtrada p'_i y q'_j con $i = 0..nDp - 1$, $j = 0..nDq - 1$.

Según el valor de dE , se aplica lo siguiente:

- 65 - Si la variable dE es igual a 2, tanto nDp como nDq se establecen iguales a 3 y se aplica el siguiente filtrado fuerte:

$$p_0' = \text{Clip3}(p_0 - 3 * t_c, p_0 + 3 * t_c, (p_2 + 2 * p_1 + 2 * p_0 + 2 * q_0 + q_1 + 4) \gg 3) \quad (8-1162)$$

$$p_1' = \text{Clip3}(p_1 - 2 * t_c, p_1 + 2 * t_c, (p_2 + p_1 + p_0 + q_0 + 2) \gg 2) \quad (8-1163)$$

5 $P_2' = \text{Clip3}(p_2 - 1 * t_c, p_2 + 1 * t_c, (2 * p_3 + 3 * P_2 + P_1 + p_0 + q_0 + 4) \gg 3) \quad (8-1164)$

$$q_0' = \text{Clip3}(q_0 - 3 * t_c, q_0 + 3 * t_c, (p_1 + 2 * p_0 + 2 * q_0 + 2 * q_1 + q_2 + 4) \gg 3) \quad (8-1165)$$

$$q_1' = \text{Clip3}(q_1 - 2 * t_c, q_1 + 2 * t_c, (p_0 + q_0 + q_1 + q_2 + 2) \gg 2) \quad (8-1166)$$

10 $q_2 = \text{Clip3}(q_2 - 1 * t_c, q_2 + 1 * t_c, (p_0 + q_0 + q_1 + 3 * q_2 + 2 * q_3 + 4) \gg 3) \quad (8-1167)$

- De lo contrario, nDp y nDq se establecen ambos iguales a 0 y se aplica el siguiente filtrado débil:

15 - Se aplica lo siguiente:

$$\Delta = (9 * (q_0 - p_0) - 3 * (q_1 - p_1) + 8) \gg 4 \quad (8-1168)$$

20 - Cuando Abs (Δ) es menor que $t_c * 10$, se aplican las siguientes etapas ordenadas:

- Los valores de muestra filtrada p_0' y q_0' se especifican del siguiente modo:

$$\Delta = \text{Clip3}(-t_c, t_c, \Delta) \quad (8-1169)$$

25 $p_0' = \text{Clip1}_\gamma(p_0 + \Delta) \quad (8-1170)$

$$q_0' = \text{Clip1}_\gamma(q_0 - \Delta) \quad (8-1171)$$

30 - Cuando dEp es igual a 1, el valor de muestra filtrada p_1' se especifica del siguiente modo:

$$\Delta p = \text{Clip3}(-(t_c \gg 1), t_c \gg 1, (((p_2 + p_0 + 1) \gg 1) - p_1 + \Delta) \gg 1) \quad (8-1172)$$

$$p_1' = \text{Clip1}_\gamma(p_1 + \Delta p) \quad (8-1173)$$

35 - Cuando dEq es igual a 1, el valor de muestra filtrada q_1' se especifica del siguiente modo:

$$\Delta q = \text{Clip3}(-(t_c \gg 1), t_c \gg 1, (((q_2 + q_0 + 1) \gg 1) - q_1 - \Delta) \gg 1) \quad (8-1174)$$

$$q_1' = \text{Clip1}_\gamma(q_1 + \Delta q) \quad (8-1175)$$

40 - nDp se establece igual a dEp + 1 y nDq se establece igual a dEq + 1.

Quando nDp es mayor que 0 y se cumplen una o más de las siguientes condiciones, nDp se establece igual a 0:

45 - `cu_transquant_bypass_flag` de la unidad de codificación que incluye el bloque de codificación que contiene la muestra p_0 es igual a 1.

50 - `pred_mode_plt_flag` de la unidad de codificación que incluye el bloque de codificación que contiene la muestra p_0 es igual a 1.

Quando nDq es mayor que 0 y se cumplen una o más de las siguientes condiciones, nDq se establece igual a 0:

55 - `cu_transquant_bypass_flag` de la unidad de codificación que incluye el bloque de codificación que contiene la muestra q_0 es igual a 1.

- `pred_mode_plt_flag` de la unidad de codificación que incluye el bloque de codificación que contiene la muestra q_0 es igual a 1.

60 8.8.3.6.7 Proceso de filtrado de una muestra de luma usando filtros largos

Las entradas de este proceso son:

- las variables `maxFilterLengthP` y `maxFilterLengthQ`,

65 - los valores de muestra p_i y q_j con $i = 0.. \text{maxFilterLengthP}$ y $j = 0.. \text{maxFilterLengthQ}$,

- las ubicaciones de p_i y q_j , (x_{P_i}, y_{P_i}) y (x_{Q_j}, y_{Q_j}) con $i = 0..maxFilterLengthP - 1$ y $j = 0..maxFilterLengthQ - 1$,
- una variable tc .

5

Las salidas de este proceso son:

- los valores de muestra filtrada p'_i y q'_j con $i = 0..maxFilterLengthP - 1$, $j = 0..maxFilterLengthQ - 1$.

10

La variable $refMiddle$ se deriva del siguiente modo:

- Si $maxFilterLengthP$ es igual a $maxFilterLengthQ$ y $maxFilterLengthP$ es igual a 5, se aplica lo siguiente:

15

$$refMiddle = (p_4 + p_3 + 2 * (p_2 + p_1 + p_0 + q_0 + q_1 + q_2) + q_3 + q_4 + 8) \gg 4 \quad (8-1176)$$

- De lo contrario, si $maxFilterLengthP$ es igual a $maxFilterLengthQ$ y $maxFilterLengthP$ no es igual a 5, se aplica lo siguiente:

20

$$refMiddle = (p_6 + p_5 + p_4 + p_3 + P_2 + p_1 + 2 * (p_0 + q_0) + q_1 + q_2 + q_3 + q_4 + q_5 + q_6 + 8) \gg 4 \quad (8-1177)$$

- De lo contrario, si se cumple una de las siguientes condiciones,

- $maxFilterLengthQ$ es igual a 7 y $maxFilterLengthP$ es igual a 5,

25

- $maxFilterLengthQ$ es igual a 5 y $maxFilterLengthP$ es igual a 7, se aplica lo siguiente:

$$refMiddle = (p_4 + p_3 + 2 * (p_2 + p_1 + p_0 + q_0 + q_1 + q_2) + q_3 + q_4 + 8) \gg 4 \quad (8-1178)$$

30

- De lo contrario, si se cumple una de las siguientes condiciones,

- $maxFilterLengthQ$ es igual a 5 y $maxFilterLengthP$ es igual a 3,

- $maxFilterLengthQ$ es igual a 3 y $maxFilterLengthP$ es igual a 5, se aplica lo siguiente:

35

$$refMiddle = (p_3 + p_2 + p_1 + p_0 + q_0 + q_1 + q_2 + q_3 + 4) \gg 3 \quad (8-1179)$$

- De lo contrario, si $maxFilterLengthQ$ es igual a 7 y $maxFilterLengthP$ es igual a 3, se aplica lo siguiente:

40

$$refMiddle = (2 * (p_2 + p_1 + p_0 + q_0) + p_0 + p_1 + q_1 + q_2 + q_3 + q_4 + q_5 + q_6 + 8) \gg 4 \quad (8-1180)$$

- De lo contrario, se aplica lo siguiente:

$$refMiddle = (p_6 + p_5 + p_4 + p_3 + p_2 + p_1 + 2 * (q_2 + q_1 + q_0 + p_0) + q_0 + q_1 + 8) \gg 4 \quad (8-1181)$$

45

Las variables $refP$ y $refQ$ se derivan del siguiente modo:

$$refP = (P_{maxFilterLengthP} + P_{maxFilterLengthP-1} + 1) \gg 1 \quad (8-1182)$$

50

$$refQ = (Q_{maxFilterLengthQ} + Q_{maxFilterLengthQ-1} + 1) \gg 1 \quad (8-1183)$$

Las variables f_i y $tcPD_i$ se definen del siguiente modo:

- Si $maxFilterLengthP$ es igual a 7, se aplica lo siguiente:

55

$$f_{0..6} = \{ 59, 50, 41, 32, 23, 14, 5 \} \quad (8-1184)$$

$$tcPD_{0..6} = \{ 6, 5, 4, 3, 2, 1, 1 \} \quad (8-1185)$$

60

- De lo contrario, si $maxFilterLengthP$ es igual a 5, se aplica lo siguiente:

$$f_{0..4} = \{ 58, 45, 32, 19, 6 \} \quad (8-1186)$$

$$tcPD_{0..4} = \{ 6, 5, 4, 3, 2 \} \quad (8-1187)$$

65

- De lo contrario, se aplica lo siguiente:

$$f_{0..2} = \{ 53, 32, 11 \} \quad (8-1188)$$

$$tcPD_{0..2} = \{ 6, 4, 2 \} \quad (8-1189)$$

5 Las variables g_j y $tcQD_j$ se definen del siguiente modo:

– Si $maxFilterLengthQ$ es igual a 7, se aplica lo siguiente:

$$10 \quad g_{0..6} = \{ 59, 50, 41, 32, 23, 14, 5 \} \quad (8-1190)$$

$$tcQD_{0..6} = \{ 6, 5, 4, 3, 2, 1, 1 \} \quad (8-1191)$$

15 – Si $maxFilterLengthQ$ es igual a 5, se aplica lo siguiente:

$$g_{0..4} = \{ 58, 45, 32, 19, 6 \} \quad (8-1192)$$

$$tcQD_{0..4} = \{ 6, 5, 4, 3, 2 \} \quad (8-1193)$$

20 – De lo contrario, se aplica lo siguiente:

$$g_{0..2} = \{ 53, 32, 11 \} \quad (8-1194)$$

$$25 \quad tcQD_{0..2} = \{ 6, 4, 2 \} \quad (8-1195)$$

Los valores de muestra filtrada p_i' y q_j' con $i = 0..maxFilterLengthP - 1$ y $j = 0..maxFilterLengthQ - 1$ se derivan del siguiente modo:

$$30 \quad p_i' = Clip3(p_i - (tc*tcPD_i) \gg 1, p_i + (tc*tcPD_i) \gg 1, (refMiddle*f_i + refP*(64 - f_i) + 32) \gg 6) \quad (8-1196)$$

$$q_j' = Clip3(q_j - (tc*tcQD_j) \gg 1, q_j + (tc*tcQD_j) \gg 1, (refMiddle*g_j + refQ*(64 - g_j) + 32) \gg 6) \quad (8-1197)$$

35 Cuando se cumplen una o más de las siguientes condiciones, el valor de muestra filtrada, p_i' , se sustituye por el valor de muestra de entrada correspondiente p_i con $i = 0..maxFilterLengthP - 1$:

– $cu_transquant_bypass_flag$ de la unidad de codificación que incluye el bloque de codificación que contiene la muestra p_i es igual a 1.

40 – $pred_mode_plt_flag$ de la unidad de codificación que incluye el bloque de codificación que contiene la muestra p_i es igual a 1.

Cuando se cumplen una o más de las siguientes condiciones, el valor de la muestra filtrada, q_j' se sustituye por el valor de muestra de entrada correspondiente q_j con $j = 0..maxFilterLengthQ - 1$:

45 – $cu_transquant_bypass_flag$ de la unidad de codificación que incluye el bloque de codificación que contiene la muestra q_j es igual a 1.

50 – $pred_mode_plt_flag$ de la unidad de codificación que incluye el bloque de codificación que contiene la muestra q_j es igual a 1.

8.8.3.6.8 Proceso de decisión para una muestra de croma

Las entradas de este proceso son:

- 55 – los valores de muestra p_0, p_3, q_0 y q_3 ,
- las variables dpq, β y tc .

60 La salida de este proceso es la variable $dSam$ que contiene una decisión.

La variable $dSam$ se especifica del siguiente modo:

- Si se cumplen todas las condiciones siguientes, $dSam$ se establece igual a 1:
- 65 – dpq es menor que $(\beta \ll 2)$,

- $Abs(p_3 - p_0) + Abs(q_0 - q_3)$ es menor que $(\beta \gg 3)$,
- $Abs(p_0 - q_0)$ es menor que $(5 * tc + 1) \gg 1$.
- De lo contrario, dSam se establece igual a 0.

//Cabe resaltar que, como se muestra en la Figura 11, la variable dSam que contiene una decisión 1107, cuando dSam se establece igual a 1, significa que el resultado de la decisión 1107 es Sí, entonces, en un ejemplo, se puede realizar la etapa 1109; cuando dSam se establece igual a 0, significa que el resultado de la decisión 1107 es NO, entonces, en un ejemplo, se puede realizar la etapa 1105.//

8.8.3.6.9 Proceso de filtrado para una muestra de croma

Este proceso solo se invoca cuando ChromaArrayType no es igual a 0.

Las entradas de este proceso son:

- la variable maxFilterLength,
- los valores de muestra de croma p_i y q_i con $i = 0..maxFilterLengthCbCr$,
- las ubicaciones de croma de p_i y q_i , (xP_i, yP_i) y (xQ_i, yQ_i) con $i = 0..maxFilterLengthCbCr - 1$,
- una variable tc.

Las salidas de este proceso son los valores de muestra filtrada p_i' y q_i' con $i = 0..maxFilterLengthCbCr - 1$.

Los valores de muestra filtrada p_i' y q_i' con $i = 0..maxFilterLengthCbCr - 1$ se derivan del siguiente modo:

- Si maxFilterLengthCbCr es igual a 3, se aplica lo siguiente:

$$p_0' = Clip3(p_0 - tc, p_0 + tc, (p_3 + p_2 + p_1 + 2 * p_0 + q_0 + q_1 + q_2 + 4) \gg 3) \quad (8-1198)$$

$$p_1' = Clip3(p_1 - tc, p_1 + tc, (2 * p_3 + p_2 + 2 * p_1 + p_0 + q_0 + q_1 + 4) \gg 3) \quad (8-1199)$$

$$p_2' = Clip3(p_2 - tc, p_2 + tc, (3 * p_3 + 2 * p_2 + p_1 + p_0 + q_0 + 4) \gg 3) \quad (8-1200)$$

$$q_0' = Clip3(q_0 - tc, q_0 + tc, (p_2 + p_1 + p_0 + 2 * q_0 + q_1 + q_2 + q_3 + 4) \gg 3) \quad (8-1201)$$

$$q_1' = Clip3(q_1 - tc, q_1 + tc, (p_1 + p_0 + q_0 + 2 * q_1 + q_2 + 2 * q_3 + 4) \gg 3) \quad (8-1202)$$

$$q_2' = Clip3(q_2 - tc, q_2 + tc, (p_0 + q_0 + q_1 + 2 * q_2 + 3 * q_3 + 4) \gg 3) \quad (8-1203)$$

- De lo contrario, se aplica el siguiente filtrado débil:

$$A = Clip3(-tc, tc, (((q_0 - p_0) << 2) + p_1 - q_1 + 4) \gg 3) \quad (8-1204)$$

$$p_0' = Clip1c(p_0 + \Delta) \quad (8-1205)$$

$$q_0' = Clip1c(q_0 - \Delta) \quad (8-1206)$$

Cuando se cumplen una o más de las siguientes condiciones, el valor de muestra filtrada, p_i' se sustituye por el valor de muestra de entrada correspondiente p_i con $i = 0..maxFilterLengthCbCr - 1$:

- `cu_transquant_bypass_flag` de la unidad de codificación que incluye el bloque de codificación que contiene la muestra p_i es igual a 1.
- `pred_mode_plt_flag` de la unidad de codificación que incluye el bloque de codificación que contiene la muestra p_i es igual a 1.

Cuando se cumplen una o más de las siguientes condiciones, el valor de muestra filtrada, q_i' , se sustituye por el valor de la muestra de entrada correspondiente q_i con $i = 0.. maxFilterLengthCbCr - 1$:

– `cu_transquant_bypass_flag` de la unidad de codificación que incluye el bloque de codificación que contiene la muestra q_i es igual a 1.

5 – `pred_mode_plt_flag` de la unidad de codificación que incluye el bloque de codificación que contiene la muestra q_i es igual a 1.

Aunque la unidad 220 de filtro de bucle se muestra en la Figura 2 como un filtro de bucle, en otras configuraciones, la unidad 220 de filtro de bucle puede implementarse como un filtro de bucle posterior. El bloque filtrado 221 también puede denominarse bloque reconstruido filtrado 221.

10 La codificación de vídeo versátil (VVC) utiliza una herramienta denominada codificación residual de croma conjunta (JCCR), que se señala en el flujo de bits usando el indicador “`tu_joint_cbc_r_residual_flag`”. Esta herramienta especifica si las muestras residuales para los dos componentes de croma Cb y Cr se codifican como un único bloque de transformada. El indicador “`tu_joint_cbc_r_residual_flag`” igual a 1 especifica que la sintaxis de la unidad de transformada incluye los niveles de coeficiente de transformada para un único bloque de transformada del que se derivan las muestras residuales tanto para Cb como para Cr. La herramienta de JCCR aprovecha el hecho de que tanto los residuos de Cb como los de Cr parecen correlacionarse en su mayoría de forma inversa entre sí.

20 Dependiendo de `tu_joint_cbc_r_residual_flag`, `tu_cbf_cb` y `tu_cbf_cr`, la variable `TuCResMode` se deriva del siguiente modo, donde `tu_cbf_cb` especifica el indicador de bloque codificado del componente Cb, `tu_cbf_cr` es el indicador de bloque codificado del componente Cr. Y `TuCResMode` indica el modo de JCCR.

- Si `tu_joint_cbc_r_residual_flag` es igual a 0, la variable `TuCResMode` se establece igual a 0;
- 25 – De lo contrario, si `tu_cbf_cb` es igual a 1 y `tu_cbf_cr` es igual a 0, la variable `TuCResMode` se establece igual a 1;
- De lo contrario, si `tu_cbf_cb` es igual a 1, la variable `TuCResMode` se establece igual a 2;
- 30 – De lo contrario, la variable `TuCResMode` se establece igual a 3.

La relación entre la “reconstrucción de los residuos de Cb y Cr” basada en las variables `tu_cbf_cb` y `tu_cbf_cr`, la variable `TuCResMode`, es la que se muestra en la tabla siguiente.

<code>tu_cbf_cb</code>	<code>tu_cbf_cr</code>	<code>TuCResMode</code>	Reconstrucción de residuos de Cb y Cr
1	0	1	$\text{resCb}[x][y] = \text{resJointC}[x][y]$ $\text{resCr}[x][y] = (\text{CSign} * \text{resJointC}[x][y]) \gg 1$
1	1	2	$\text{resCb}[x][y] = \text{resJointC}[x][y]$ $\text{resCr}[x][y] = \text{CSign} * \text{resJointC}[x][y]$
0	1	3	$\text{resCb}[x][y] = (\text{CSign} * \text{resJointC}[x][y]) \gg 1$ $\text{resCr}[x][y] = \text{resJointC}[x][y]$

45 La variable `CSgin` es un valor de signo (+1 o -1) y esto se indica en el encabezado del segmento. `resJointC[x][y]` es el residuo real transmitido en el flujo de bits.

`resCb[x][y]` indica las muestras residuales derivadas para el componente de croma Cb;

50 `resCr[x][y]` indica las muestras residuales derivadas para el componente de croma Cr.

La presente descripción puede usar las tablas de mapeo de QP de croma respectivas para cada uno de los componentes de croma Cb y Cr y los residuos de Cb-Cr conjuntos. Cuando el elemento de sintaxis “`same_qp_table_for_chroma`” es igual a 1, especifica que se utilizan las mismas tablas de QP de croma y que estas mismas tablas se aplican a los residuos de Cb, Cr y Cb-Cr conjuntos. Cuando “`same_qp_table_for_chroma`” es igual a 0, representa que se utiliza una tabla de mapeo de QP de croma diferente para los residuos de Cb, Cr o Cb-Cr conjuntos. Las tres tablas de mapeo de croma pueden representarse de diferentes formas.

60 Como se muestra en las Figuras 12A y 12B, hay tres subtablas (es decir, las filas con un valor de índice igual a 0, 1 y 2). Como el contenido de la fila 2 es diferente del de la fila 0 y/o 1, el elemento de sintaxis `same_qp_table_for_chroma` debe ser igual a 0. De lo contrario, si el elemento de sintaxis `same_qp_table_for_chroma` es igual a 1, el contenido de las filas 1 y 2 será el mismo que el de la fila 0.

65 Como se muestra en la Figura 13, hay tres tablas separadas, diseñadas para los residuos de Cb, Cr y Cb-Cr conjuntos, respectivamente. Como el contenido de las tablas para los residuos de Cb, Cr y Cb-Cr conjuntos es diferente entre sí, el elemento de sintaxis `same_qp_table_for_chroma` debe ser igual a 0. De lo contrario, si el

elemento de sintaxis `same_qp_table_for_chroma` es igual a 1, el contenido de las tablas para los residuos de Cb, Cr y Cb-Cr conjuntos debería ser el mismo.

5 Los elementos de sintaxis `num_points_in_qp_table_minus1 [i]`, `delta_qp_in_val_minus1 [i] [j]`, `delta_qp_out_val [i] [j]` se utilizan además para derivar las tablas de mapeo de QP de croma. La semántica de estos elementos de sintaxis y el procedimiento para derivar las tablas de mapeo de QP de croma son los que se muestran a continuación:

10 `num_points_in_qp_table_minus1 [i]` plus 1 especifica el número de puntos utilizados para describir la i-ésima tabla de mapeo de QP de croma. El valor de `num_points_in_qp_table_minus1 [i]` estará en el intervalo de 0 a 63 + `QpBdOffsetc`, ambos inclusive. Cuando `num_points_in_qp_table_minus1 [0]` no está presente en el flujo de bits, se deduce que el valor de `num_points_in_qp_table_minus1 [0]` es igual a 0.

15 `delta_qp_in_val_minus1 [i] [j]` especifica un valor de delta que se utiliza para derivar la coordenada de entrada del j-ésimo punto de pivote de la tabla de mapeo de QP de croma. Cuando `delta_qp_in_val_minus1 [0] [j]` no está presente en el flujo de bits, se deduce que el valor de `delta_qp_in_val_minus1 [0] [j]` es igual a 0.

20 `delta_qp_out_val [i] [j]` especifica un valor de delta que se utiliza para derivar la coordenada de salida del i-ésimo punto de pivote de la tabla de mapeo de QP de croma. Cuando `delta_qp_out_val [0] [j]` no está presente en el flujo de bits, se deduce que el valor de `delta_qp_out_val [0] [j]` es igual a 0.

Cabe señalar que en la presente memoria, los detalles del ejemplo sobre cómo derivar la primera tabla de mapeo de QP de croma, la segunda tabla de mapeo de QP de croma y la tercera tabla de mapeo de QP de croma se describirán a continuación.

25 La i-ésima tabla de mapeo de QP de croma `ChromaQpTable [i]` para $i = 0..same_qp_table_for_chroma? 0: 2$ se obtiene del siguiente modo:

```

qplnVal[ i ][ 0 ] = -QpBdOffsetc + delta_qp_in_val_minus1[ i ][ 0 ]
30 qpOutVal[ i ][ 0 ] = -QpBdOffsetc + delta_qp_out_val[ i ][ 0 ]
for( j = 1; j <= num_points_in_qp_table_minus1[ i ]; j++ ) {
qplnVal[ i ][ j ] = qplnVal[ i ][ j - 1 ] + delta_qp_in_val_minus1 [ i ][ j ] + 1
35 qpOutVal[ i ][ j ] = qpOutVal[ i ][ j - 1 ] + delta_qp_out_val[ i ][ j ] }
ChromaQpTable[ i ][ qplnVal[ i ][ 0 ] ] = qpOutVal[ i ][ 0 ]
40 para (k = qplnVal[ i ][ 0 ] - 1; k >= -QpBdOffsetc; k -- )
ChromaQpTable[ i ][ k ] = Clip3( -QpBdOffsetc, 63,
ChromaQpTable[ i ][ k + 1 ] - 1 ) (7-31)
45 for( j = 0; j < num_points_in_qp_table_minus1[ i ]; j++ ) {
sh = ( delta_qp_in_val_minus1 [ i ][ j + 1 ] + 2 ) >> 1
50 para ( k = qplnVal[ i ][ j ] + 1, m = 1; k <= qplnVal[ i ][ j + 1 ]; k++, m++)
ChromaQpTable[ i ][ k ] = ChromaQpTable[ i ][ qplnVal[ i ][ j ] ] +
55 (delta_qp_out_val[ i ][ j + 1 ] * m + sh) /
(delta_qp_in_val_minus1 [i] [j + 1] + 1)
}
60 para( k = qplnVal[ i ][ num_points_in_qp_table_minus1[ i ] ] + 1; k <= 63; k++ )
ChromaQpTable[ i ][ k ] = Clip3(-QpBdOffsetc, 63, ChromaQpTable[ i ][ k - 1 ] + 1)
65 Cuando same_qp_table_for_chroma es igual a 1, ChromaQpTable [1] [k] y ChromaQpTable [2] [k] se establecen
iguales a ChromaQpTable [0] [k] para  $k = -QpBdOffsetc.. 63$ .
```

Es un requisito para la conformidad del flujo de bits que los valores de $qpInVal [i][j]$ y $qpOutVal [i][j]$ estén en el intervalo de $-QpBdOffset_c$ a 63, ambos inclusive para $i = 0..same_qp_table_for_chroma? 0: 2$ y $j = 0..num_points_in_qp_table_minus1 [i]$.

5 Téngase en cuenta que la tabla ChormaQPmapping también se puede expresar usando una fórmula simple que toma como entrada el valor de QP de luma (QP_l) y el valor del componente de color ($cldx$) y, a continuación, genera el valor de Qp de croma correspondiente (QP_c). La fórmula puede representar una relación lineal entre el QP de luma y el QP de croma. Por ejemplo, la fórmula puede ser la siguiente: $QP_c = QP_l - x$, donde x es constante y depende del valor del componente de color ($cldx$) y x puede tomar diferentes valores para diferentes índices de componente de color, incluido el componente Cb-Cr conjunto.

15 Las realizaciones del codificador 20 de vídeo (unidad 220 de filtro de bucle respectivamente) pueden configurarse para dar salida a parámetros de filtro de bucle (tales como parámetros de filtro de SAO o parámetros de filtro de ALF o parámetros de LMCS), p. ej., directamente o codificados a través de la unidad 270 de codificación entrópica, para que, p. ej., un decodificador 30 pueda recibir y aplicar los mismos parámetros de filtro de bucle o filtros de bucle respectivos para la decodificación.

Memoria intermedia de imágenes descodificadas

20 La memoria 230 intermedia de imágenes decodificadas (DPB) puede ser una memoria que almacena imágenes de referencia o, en general, datos de imagen de referencia para la codificación de datos de vídeo por parte del codificador 20 de vídeo. La memoria DPB 230 puede estar formada por cualquiera de una variedad de dispositivos de memoria, tales como una memoria de acceso aleatorio dinámica (DRAM), incluida una DRAM síncrona (SDRAM), una RAM magnetorresistiva (MRAM), una RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. La memoria 230 intermedia de imágenes decodificadas (DPB) puede configurarse para almacenar uno o más bloques filtrados 221. La memoria 230 intermedia de imágenes decodificadas puede configurarse además para almacenar otros bloques previamente filtrados, p. ej., unos bloques 221 previamente reconstruidos y filtrados, de la misma imagen actual o de distintas imágenes, p. ej., unas imágenes previamente reconstruidas, y puede proporcionar unas imágenes completas previamente reconstruidas, es decir, decodificadas, (y unos bloques de referencia y muestras correspondientes) y/o una imagen actual parcialmente reconstruida (y unos bloques de referencia y muestras correspondientes) para, por ejemplo, la interpredicción. La memoria 230 intermedia de imágenes decodificadas (DPB) también puede configurarse para almacenar uno o más bloques reconstruidos no filtrados 215 o, en general, unas muestras reconstruidas no filtradas, p. ej., si el bloque reconstruido 215 no es filtrado por la unidad 220 de filtro de bucle, o cualquier otra versión más procesada de los bloques o muestras reconstruidos.

35 Selección de modo (partición y predicción)

La unidad 260 de selección de modo comprende la unidad 262 de partición, la unidad 244 de interpredicción y la unidad 254 de intrapredicción y está configurada para recibir u obtener datos de imagen originales, p. ej., un bloque original 203 (un bloque 203 actual de la imagen 17 actual), y datos de imagen reconstruidos, p. ej., unas muestras reconstruidas filtradas y/o no filtradas o unos bloques de la misma imagen (actual) y/o de una o una pluralidad de imágenes previamente decodificadas, p. ej., procedentes de la memoria 230 intermedia de imágenes decodificadas o de otras memorias intermedias (p. ej., una memoria intermedia de línea, que no se muestra). Los datos de imagen reconstruidos se usan como datos de imagen de referencia para la predicción, p. ej., la interpredicción o la intrapredicción, para obtener un bloque 265 de predicción o predictor 265.

45 La unidad 260 de selección de modo puede configurarse para determinar o seleccionar una partición para un modo de predicción de bloque actual (que no incluye ninguna partición) y un modo de predicción (p. ej., un modo de intra o interpredicción) y generar un bloque 265 de predicción correspondiente, el cual se usa para el cálculo del bloque residual 205 y para la reconstrucción del bloque reconstruido 215. Algunas realizaciones de la unidad 260 de selección de modo pueden configurarse para seleccionar la partición y el modo de predicción (p. ej., de entre los admitidos por o los disponibles para la unidad 260 de selección de modo) que proporcionan la mejor coincidencia o, en otras palabras, el residuo mínimo (un residuo mínimo da como resultado una mejor compresión para la transmisión o el almacenamiento) o una sobrecarga de señalización mínima (una sobrecarga de señalización mínima da como resultado una mejor compresión para la transmisión o el almacenamiento) o que consideran o equilibran ambos. La unidad 260 de selección de modo puede configurarse para determinar la partición y el modo de predicción basándose en una optimización de distorsión de velocidad (RDO, por sus siglas en inglés), es decir, para seleccionar el modo de predicción que proporciona una distorsión de velocidad mínima. Términos como “mejor”, “mínimo”, “óptimo”, etc. en este contexto no se refieren necesariamente a un “mejor” general, “mínimo”, “óptimo”, etc., pero también pueden referirse al cumplimiento de un criterio de terminación o selección como un valor que excede o cae por debajo de un umbral u otras restricciones que conducen potencialmente a una “selección subóptima” pero que reduce la complejidad y el tiempo de procesamiento.

65 En otras palabras, la unidad 262 de partición puede configurarse para dividir una figura de una secuencia de vídeo en una secuencia de unidades de árbol de codificación (CTU,), y la CTU 203 puede dividirse además en particiones de bloque más pequeñas o subbloques (que a su vez forman bloques), p. ej., de modo iterativo usando una partición de árbol cuaternario (QT, en inglés), una partición binaria (BT, en inglés) o una partición de árbol ternario (TT, en inglés) o cualquier combinación de las mismas, y para realizar, p. ej., la predicción para cada una de las particiones

de bloque o subbloques, en donde la selección de modo comprende la selección de la estructura de árbol del bloque dividido 203 y los modos de predicción se aplican a cada una de las particiones de bloque o subbloques.

5 A continuación se explicará más detalladamente la partición (p. ej., por parte de la unidad 260 de partición) y el procesamiento predictivo (por parte de la unidad 244 de interpredicción y la unidad 254 de intrapredicción) realizados por un codificador 20 de vídeo de ejemplo.

Partición

10 La unidad 262 de partición puede configurarse para dividir una imagen de una secuencia de vídeo en una secuencia de unidades de árbol de codificación (CTU), y la unidad 262 de partición puede dividir (o particionar) una unidad 203 de árbol de codificación (CTU) en particiones más pequeñas, p. ej., bloques cuadrados o rectangulares de tamaño más pequeño. Para una imagen que tiene tres matrices de muestra, una CTU consiste en un bloque $N \times N$ de muestras de luma junto con dos bloques correspondientes de muestras de croma. El tamaño máximo permitido del bloque de luma en una CTU se especifica en 128×128 en la codificación de vídeo versátil (VVC) en desarrollo, pero en el futuro se podrá especificar como un valor en lugar de 128×128 , por ejemplo, 256×256 . Las CTU de una imagen pueden agruparse como grupos de segmentos/mosaicos, mosaicos o ladrillos. Un mosaico cubre una región rectangular de una imagen y un mosaico se puede dividir en uno o más ladrillos. Un ladrillo consiste en varias filas de CTU dentro de un mosaico. Un mosaico que no se divide en varios ladrillos puede denominarse ladrillo. Sin embargo, un ladrillo es un verdadero subconjunto de un mosaico y no se denomina mosaico. La VVC admite dos modos de grupos de mosaico, a saber, el modo de grupo de segmento/mosaico con escaneo ráster y el modo de segmento rectangular. En el modo de grupo de mosaicos con escaneo ráster, un grupo de segmentos/mosaicos contiene una secuencia de mosaicos en el escaneo ráster de mosaicos de una imagen. En el modo de segmento rectangular, un segmento contiene varios ladrillos de una imagen que, en conjunto, forman una región rectangular de la imagen. Los ladrillos dentro de un segmento rectangular están en el orden del escaneo rasterizado de ladrillos del segmento. Estos bloques más pequeños (que también pueden denominarse subbloques) pueden partirse más en particiones aún más pequeñas. Esto se denomina también partición en árbol o partición en árbol jerárquico, en la que un bloque raíz, p. ej., en un nivel de árbol raíz 0 (nivel jerárquico 0, profundidad 0), puede partirse recursivamente, p. ej., partirse en dos o más bloques de un siguiente nivel de árbol inferior, p. ej., unos nodos en un nivel de árbol 1 (nivel jerárquico 1, profundidad 1), en la que estos bloques pueden partirse de nuevo en dos o más bloques de un nivel inferior siguiente, p. ej., un nivel de árbol 2 (nivel jerárquico 2, profundidad 2), etc. hasta que se pone fin a la partición, p. ej., porque se cumple un criterio de finalización, p. ej., se alcanza una profundidad de árbol máxima o un tamaño de bloque mínimo. Los bloques que no se parten mas también se denominan bloques hoja o nodos de hoja del árbol. Un árbol que usa una partición en dos particiones se denomina árbol binario (BT), un árbol que usa una partición en tres particiones se denomina árbol ternario (TT) y un árbol que usa una partición en cuatro particiones se denomina árbol cuádruple (QT).

Por ejemplo, una unidad de árbol de codificación (CTU) puede ser o comprender un bloque CTB de muestras de luma, dos bloques CTB correspondientes de muestras de croma de una imagen que tiene tres matrices de muestras o un bloque CTB de muestras de una imagen monocromática o de una imagen que se codifica usando tres planos de color distintos y unas estructuras sintácticas utilizadas para codificar las muestras. En consecuencia, un bloque de árbol de codificación (CTB) puede ser un bloque de $N \times N$ de muestras para cierto valor de N , de manera que la división de un componente en bloques CTB es una partición. Una unidad de codificación (CU) puede ser o comprender un bloque de codificación de muestras de luma, dos bloques de codificación correspondientes de muestras de croma de una imagen que tiene tres matrices de muestras o un bloque de codificación de muestras de una imagen monocromática o de una imagen que se codifica usando tres planos de color distintos y unas estructuras sintácticas utilizadas para codificar las muestras. En consecuencia, un bloque de codificación (CB) puede ser un bloque de $M \times N$ de muestras para algunos valores de M y N , de manera que la división de un bloque CTB en bloques de codificación es una partición.

50 En algunas realizaciones, p. ej., según la codificación HEVC, una unidad de árbol de codificación (CTU) puede dividirse en unidades CU usando una estructura de árbol cuádruple denotada como árbol de codificación. La decisión de codificar un área de imagen usando la predicción interimagen (temporal) o intrainimagen (espacial) se realiza a nivel de CU de hoja. Cada CU de hoja puede dividirse adicionalmente en una, dos o cuatro PU según el tipo de división de PU. Dentro de una PU, se aplica el mismo proceso de predicción y la información relevante se transmite al descodificador en una base de PU. Después de obtener el bloque residual aplicando el proceso de predicción basado en el tipo de división de PU, una CU de hoja puede dividirse en unidades de transformada (TU) según otra estructura de árbol cuaternario similar al árbol de codificación para la CU.

60 En realizaciones, por ejemplo, según la última norma de codificación de vídeo actualmente en desarrollo, que se denomina codificación de vídeo versátil (VVC), un árbol de tipo múltiple anidado de árbol cuaternario combinado usando una estructura de segmentación en divisiones binarias y ternarias, se usa, por ejemplo, para dividir una unidad de árbol de codificación. En la estructura de árbol de codificación dentro de una unidad de árbol de codificación, una CU puede tener una forma cuadrada o rectangular. Por ejemplo, la unidad de árbol de codificación (CTU) se divide primero por un árbol cuaternario. Entonces los nodos de hoja de árbol cuaternario pueden dividirse adicionalmente mediante una estructura de árbol de tipo múltiple. Hay cuatro tipos de división en la estructura de árbol de tipo múltiple, la división binaria vertical (SPLIT_BT_VER), la división binaria horizontal (SPLIT_BT_HOR), la división ternaria vertical (SPLIT_TT_VER), y la división ternaria horizontal (SPLIT_TT_HOR). Los nodos de hoja de árbol de tipo múltiple se denominan unidades de

codificación (CU), y a menos que la CU sea demasiado grande para la longitud de transformada máxima, esta segmentación se usa para el procesamiento de predicción y transformación sin ninguna partición adicional. Esto significa que, en la mayoría de los casos, la CU, PU y TU tienen el mismo tamaño de bloque en el árbol cuaternario con una estructura de bloque de codificación de árbol de tipo múltiple anidada. La excepción se produce cuando una longitud de transformada máxima soportada es menor que la anchura o altura del componente de color de la CU. La VVC desarrolla un mecanismo de señalización único de la partición que divide la información en árbol cuaternario con una estructura de árbol de codificación de tipo múltiple anidada. En el mecanismo de señalización, una unidad de árbol de codificación (CTU) se trata como la raíz de un árbol cuaternario y se divide primero por una estructura de árbol cuaternario. Cada nodo de hoja de árbol cuaternario (cuando lo suficientemente grande para permitirle) se divide adicionalmente por una estructura de árbol de tipo múltiple. En la estructura de árbol de tipo múltiple, se señala un primer indicador (mtt_splid_cu_flag) para indicar si el nodo está partido; cuando un nodo se divide adicionalmente, un segundo indicador (mtt_splifit_cu_vertical_flag) se señala para indicar la dirección de división y, a continuación, se señala un tercer indicador (mtt_splid_cu_binary_flag) para indicar si la división es una división binaria o una división ternaria. Basado en los valores de mtt_splifit_cu_vertical_flag y mtt_splifit_cu_binario_flag, el modo de división de árbol de tipo múltiple (MttSplitMode) de una CU puede derivarse mediante un decodificador basado en una regla o una tabla predefinidas. Cabe señalar, para un cierto diseño, por ejemplo, 64 x 64 bloques luma y 32 x 32 de diseño de pipeta croma en decodificadores de hardware VVC, la división de TT está prohibida cuando ya sea ancho o altura de un bloque de codificación de luma es mayor que 64, como se muestra en la Figura 6. La división de TT también está prohibida cuando la anchura o altura de un bloque de codificación de croma es mayor que 32. El diseño de canalización dividirá una imagen en unidades de datos de tuberías virtuales (PPDU) que se definen como unidades no superpuestas en una imagen. En decodificadores de hardware, las VPDU sucesivas pueden procesarse a través de múltiples etapas de canalización simultáneamente. El tamaño de VPDU es aproximadamente proporcional al tamaño de la memoria intermedia en la mayoría de las etapas de la tubería, por lo que es importante mantener el tamaño de VPDU pequeño. En la mayoría de los decodificadores de hardware, el tamaño de VPDU puede establecerse en el tamaño máximo de bloque de transformada (TB). Sin embargo, en la VVC, la partición de árbol ternario (TT) y árbol binario (BT) puede conducir a un aumento de tamaño de las VPDU.

Además, debe observarse que, cuando una parte de un bloque de nodo de árbol excede el límite de imagen inferior o derecha, el bloque de nodo de árbol es forzado a dividirse hasta que todas las muestras de cada CU codificada están ubicadas dentro de los límites de imagen.

Como ejemplo, la herramienta de intrasubpartición (ISP) puede dividir bloques intrapredichos de luma en vertical o en horizontal en 2 o 4 subdivisiones dependiendo del tamaño de bloque.

En un ejemplo, la unidad 260 de selección de modo del codificador 20 de vídeo puede configurarse para realizar cualquier combinación de las tecnologías de partición descritas en la presente memoria.

Tal y como se ha descrito anteriormente, el codificador 20 de vídeo se configura para determinar o seleccionar el mejor modo de predicción o un modo de predicción óptimo de un conjunto de modos de predicción (p. ej., predeterminados). El conjunto de modos de predicción puede comprender, p. ej., modos de intrapredicción y/o modos de interpredicción.

Intrapredicción

El conjunto de modos de intrapredicción puede comprender 35 modos de intrapredicción diferentes, p. ej. modos no direccionales como DC (o media) y modo plano, o modos direccionales, por ejemplo, como se define en HEVC, o puede comprender 67 modos de intrapredicción diferentes, por ejemplo, modos no direccionales como modo DC (o media) y modo plano, o modos direccionales, por ejemplo, como se define para VVC. Como ejemplo, varios modos de intrapredicción angular convencionales son reemplazados de manera adaptativa por modos de intrapredicción de ángulo amplio para los bloques no cuadrados, p. ej., como se define en VVC. Como otro ejemplo, para evitar operaciones de división para la predicción de DC, solo el lado más largo se usa para calcular el promedio para bloques no cuadrados. Y, los resultados de la intrapredicción del modo plano pueden modificarse adicionalmente mediante un método de combinación de intrapredicción dependiente de la posición (PDPC).

La unidad 254 de intrapredicción está configurada para usar unas muestras reconstruidas de unos bloques vecinos de la misma imagen actual para generar un bloque 265 de intrapredicción según un modo de intrapredicción del conjunto de modos de intrapredicción.

La unidad 254 de intrapredicción (o, en general, la unidad 260 de selección de modo) está configurada además para dar salida a unos parámetros de intrapredicción (o, en general, a información indicativa del modo de intrapredicción seleccionado para el bloque) hacia la unidad 270 de codificación entrópica en forma de unos elementos sintácticos 266 para su inclusión en los datos 21 de imagen codificados a fin de que, p. ej., el decodificador 30 de vídeo pueda recibir y utilizar los parámetros de predicción para la decodificación.

Interpredicción

El conjunto de (o posibles) modos de interpredicción depende de las imágenes de referencia disponibles (es decir, imágenes anteriores al menos parcialmente decodificadas, p. ej., almacenadas en DBP 230) y otros parámetros de interpredicción,

p.ej., si se utiliza toda la imagen de referencia o solo una parte, p. ej., un área de ventana de búsqueda alrededor del área del bloque actual, si se utiliza la imagen de referencia para buscar un bloque de referencia que coincida mejor, y/o, p. ej., si se aplica la interpolación de píxeles, p. ej., la interpolación de medio/semipíxel, un cuarto de píxel y/o 1/16 de píxel, o no.

5 Además de los modos de predicción anteriores, se pueden aplicar modo de omisión, modo directo y/u otros modos de interpredicción.

10 Por ejemplo, para la predicción de fusión extendida, la lista de candidatos de fusión de dicho modo se construye incluyendo los siguientes cinco tipos de candidatos en orden: MVP espacial de CU vecinas espaciales, MVP Temporal de CU coladas, MVP basada en el historial desde una tabla FIFO, MVP promedio por pares y los MV cero. Y se puede aplicar un refinamiento del vector de movimiento del lado del descodificador basado en dos lados (DMVR) para aumentar la precisión de los MV del modo de fusión. Modo fusión con MVD (MMVD), que proviene del modo de fusión con diferencias de vectores de movimiento. Un indicador MMVD se señala justo después de enviar un indicador de omisión y un indicador de fusión para especificar si el modo MMVD se utiliza para una CU. Y se puede aplicar un esquema de resolución de vector de movimiento adaptativo a nivel de CU (AMVR). AMVR permite codificar MVD de la CU en diferentes precisiones. Dependiendo del modo de predicción para la CU actual, los MVD de la CU actual pueden seleccionarse adaptativamente. Cuando una CU está codificada en modo de fusión, el modo de inter/intrapredicción combinado (CIIP) puede aplicarse a la CU actual. El promedio ponderado de las señales interpredicción e intrapredicción se realiza para obtener la predicción de CIIP. Predicción con compensación de movimiento afín, el campo de movimiento de afín del bloque se describe mediante información de movimiento de dos vectores de control (4 parámetros) o tres vectores de movimiento de punto de control (6 parámetros). Predicción de vector de movimiento temporal basada en subbloque (SbTMVP), que es similar a la predicción de vector de movimiento temporal (TMVP) en HEVC, pero predice los vectores de movimiento de las sub-CU dentro de la CU actual. El flujo óptico bidireccional (BDF), denominado previamente BIO, es una versión más simple que requiere mucho menos cálculo, especialmente en términos de número de multiplicaciones y el tamaño del multiplicador. Modo de partición triángulo, en tal modo, una CU se divide uniformemente en dos particiones en forma de triángulo, usando la división diagonal o la división antidiagonal. Además, el modo de bipredicción se extiende más allá del promedio simple para permitir el promedio ponderado de las dos señales de predicción.

30 La unidad 244 de interpredicción puede incluir una unidad de estimación de movimiento (ME, por sus siglas en inglés) y una unidad de compensación de movimiento (MC, por sus siglas en inglés) (en la Figura 2 no se muestra ninguna de las dos). La unidad de estimación de movimiento puede configurarse para recibir u obtener el bloque 203 de imagen (el bloque 203 de imagen actual de la imagen 17 actual) y una imagen decodificada 231, o al menos uno o una pluralidad de bloques previamente reconstruidos, p.ej., unos bloques reconstruidos de una o una pluralidad de otras/distintas imágenes 231 previamente decodificadas, para una estimación de movimiento. P. ej., una secuencia de vídeo puede comprender la imagen actual y las imágenes 231 previamente decodificadas o, en otras palabras, la imagen actual y las imágenes 231 previamente decodificadas pueden ser parte de o formar una secuencia de imágenes que forman una secuencia de vídeo.

40 El codificador 20 puede estar configurado, p. ej., para seleccionar un bloque de referencia de entre una pluralidad de bloques de referencia de las mismas o distintas imágenes de la pluralidad de otras imágenes y proporcionar una imagen de referencia (o un índice de imagen de referencia) y/o un desplazamiento (un desplazamiento espacial) entre la posición (unas coordenadas x, y) del bloque de referencia y la posición del bloque actual como un(os) parámetro(s) de interpredicción a la unidad de estimación de movimiento. Este desplazamiento también se denomina vector de movimiento (MV).

45 La unidad de compensación de movimiento está configurada para obtener, p. ej., recibir, un parámetro de interpredicción y realizar una interpredicción basándose en o usando el parámetro de interpredicción para obtener un bloque 265 de interpredicción. La compensación de movimiento, realizada por la unidad de compensación de movimiento, puede implicar grabar o generar el bloque de predicción basándose en el movimiento/vector de bloque determinado por la estimación de movimiento, realizando posiblemente interpolaciones a la precisión de subpíxel. El filtrado de interpolación puede generar muestras de píxeles adicionales a partir de muestras de píxeles conocidas, aumentando así potencialmente el número de bloques de predicción candidatos que pueden usarse para codificar un bloque de imagen. Tras recibir el vector de movimiento para la PU del bloque de imagen actual, la unidad de compensación de movimiento puede localizar el bloque de predicción al que los puntos de vector de movimiento en una de las listas de imágenes de referencia.

55 La unidad de compensación de movimiento también puede generar elementos sintácticos asociados con los bloques y segmentos de vídeo para su uso por el descodificador 30 de vídeo al descodificar los bloques de imagen del fragmento de vídeo. Además o como alternativa a cortes y elementos de sintaxis respectivos, los grupos de baldosas y/o baldosas y los elementos de sintaxis respectivos pueden generarse o usarse.

60 Codificación entrópica

65 La unidad 270 de codificación entrópica está configurada para aplicar, por ejemplo, un plan o algoritmo de codificación entrópica (p. ej., un plan de codificación de longitud variable [VLC, por sus siglas en inglés], un plan VLC adaptativo por contexto [CAVLC, por sus siglas en inglés], un plan de codificación aritmética, una binarización, una codificación aritmética binaria adaptativa al contexto [CABAC, por sus siglas en inglés], una codificación aritmética binaria adaptativa al contexto basada en la sintaxis [SBAC, por sus siglas en inglés], una codificación entrópica por partición en intervalos probabilísticos [PIPE, por sus siglas en inglés] u otra metodología o técnica de codificación entrópica) u obviar (ninguna compresión) los

coeficientes cuantificados 209, los parámetros de interpredicción, los parámetros de intrapredicción, los parámetros de filtro de bucle y/u otros elementos sintácticos para obtener unos datos 21 de imagen codificados a los que puede darse salida por la salida 272, p. ej., en forma de un flujo 21 de bits codificado, para que, p. ej., el decodificador 30 de vídeo pueda recibir y usar los parámetros para la decodificación. El flujo 21 de bits codificado puede transmitirse al decodificador 30 de vídeo o archivar-se para su transmisión posterior al o su recuperación por parte del decodificador 30 de vídeo.

Para codificar un flujo de vídeo pueden usarse otras variaciones estructurales del codificador 20 de vídeo. Por ejemplo, en el caso de ciertos bloques o tramas, un decodificador 20 no basado en transformadas puede cuantificar la señal residual directamente sin necesitar la unidad 206 de procesamiento de transformada. En otra implementación, un codificador 20 puede tener la unidad 208 de cuantificación y la unidad 210 de cuantificación inversa combinadas en una sola unidad.

Decodificador y método de decodificación

La Figura 3 muestra un decodificador 30 de vídeo que está configurado para implementar las técnicas de esta presente solicitud. El decodificador 30 de vídeo está configurado para recibir unos datos 21 de imagen codificados (p. ej., un flujo 21 de bits codificado), p. ej., codificados por el codificador 20, para obtener una imagen decodificada 331. Los datos de imagen codificados o flujo de bits codificados comprende(n) información para decodificar los datos de imagen codificados, p. ej., datos que representan bloques de imagen de un segmento de vídeo codificado (y/o mosaicos o grupos de mosaicos) y elementos de sintaxis asociados.

En el ejemplo de la Figura 3, el decodificador 30 comprende una unidad 304 de decodificación entrópica, una unidad 310 de cuantificación inversa, una unidad 312 de procesamiento de transformada inversa, una unidad 314 de reconstrucción (p. ej., un sumador 314), un filtro 320 de bucle, una memoria 330 intermedia de imágenes decodificadas (DBP, en inglés), una unidad 360 de aplicación de modo, una unidad 344 de interpredicción y una unidad 354 de intrapredicción. La unidad 344 de interpredicción puede ser o incluir una unidad de compensación de movimiento. El decodificador 30 de vídeo puede, en algunos ejemplos, realizar un paso de descodificación generalmente recíproco al paso de codificación descrito con respecto al codificador 100 de vídeo de la Figura 2.

Tal y como se explicó con respecto al codificador 20, se dice también que la unidad 210 de cuantificación inversa, la unidad 212 de procesamiento de transformada inversa, la unidad 214 de reconstrucción, el filtro 220 de bucle, la memoria 230 intermedia de imágenes decodificadas (DPB), la unidad 344 de interpredicción y la unidad 354 de intrapredicción forman el "decodificador incorporado" del codificador 20 de vídeo. Por consiguiente, la unidad 310 de cuantificación inversa puede ser idéntica en función a la unidad 110 de cuantificación inversa, la unidad 312 de procesamiento de transformada inversa puede ser idéntica en función a la unidad 212 de procesamiento de transformada inversa, la unidad 314 de reconstrucción puede ser idéntica en función a la unidad 214 de reconstrucción, el filtro 320 de bucle puede ser idéntico en función al filtro 220 de bucle y la memoria 330 intermedia de imágenes decodificadas puede ser idéntica en función a la memoria 230 intermedia de imágenes decodificadas. Por lo tanto, las explicaciones proporcionadas para las respectivas unidades y funciones del codificador 20 de vídeo aplican correspondientemente a las respectivas unidades y funciones del decodificador 30 de vídeo.

Decodificación entrópica

La unidad 304 de decodificación entrópica está configurada para analizar el flujo 21 de bits (o, en general, los datos 21 de imagen codificados) y realizar, por ejemplo, una decodificación entrópica en los datos 21 de imagen codificados para obtener, p. ej., unos coeficientes cuantificados 309 y/o unos parámetros de codificación decodificados (no mostrados en la Figura 3), p. ej., cualquiera de o todos los parámetros de interpredicción (p. ej., el índice de imagen de referencia y el vector de movimiento), los parámetros de intrapredicción (p. ej., el modo o el índice de intrapredicción), los parámetros de transformada, los parámetros de cuantificación, los parámetros de filtro de bucle y/u otros elementos sintácticos. La unidad 304 de decodificación entrópica puede configurarse para aplicar los algoritmos o planes de decodificación que corresponden a los planes de codificación que se han descrito con respecto a la unidad 270 de codificación entrópica del codificador 20. La unidad 304 de decodificación entrópica puede configurarse además para proporcionar los parámetros de interpredicción, los parámetros de intrapredicción y/u otros elementos de sintaxis a la unidad 360 de aplicación de modo y otros parámetros a otras unidades del decodificador 30. El decodificador 30 de vídeo puede recibir los elementos de sintaxis a nivel del segmento de vídeo y/o el nivel del bloque de vídeo. Además o como alternativa a los segmentos y respectivos elementos de sintaxis, se pueden recibir y/o usar grupos de mosaicos y/o mosaicos y los respectivos elementos de sintaxis.

Cuantificación inversa

La unidad 310 de cuantificación inversa puede configurarse para recibir unos parámetros de cuantificación (QP) (o, en general, información relacionada con la cuantificación inversa) y unos coeficientes cuantificados de los datos 21 de imagen codificados (p. ej., mediante un análisis y/o una decodificación por parte de, p. ej., la unidad 304 de decodificación entrópica) y para aplicar, basándose en los parámetros de cuantificación, una cuantificación inversa a los coeficientes cuantificados decodificados 309 a fin de obtener unos coeficientes descuantificados 311, que también pueden denominarse coeficientes 311 de transformada. El proceso de cuantificación inversa puede incluir el uso de un parámetro de cuantificación determinado por el codificador 20 de vídeo para cada bloque de vídeo en el segmento de vídeo (o mosaico o grupo de mosaico) para determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa que debería aplicarse.

Transformada inversa

La unidad 312 de procesamiento de transformada inversa puede configurarse para recibir unos coeficientes descuantificados 311, denominados también coeficientes 311 de transformada, y para aplicar una transformada a los coeficientes descuantificados 311 a fin de obtener unos bloques residuales 213 reconstruidos en el dominio de muestra. Los bloques residuales 213 reconstruidos también pueden denominarse bloques 313 de transformada. La transformada puede ser una transformada inversa, p. ej., una transformada DCT inversa, una transformada DST inversa, una transformada entera inversa o un proceso de transformada inversa que sea conceptualmente similar. La unidad 312 de procesamiento de transformada inversa puede configurarse además para recibir unos parámetros de transformada o una correspondiente información de los datos 21 de imagen codificados (p. ej., mediante un análisis y/o una decodificación, p. ej., por parte de la unidad 304 de decodificación entrópica) para determinar la transformada que hay que aplicar a los coeficientes descuantificados 311.

Reconstrucción

La unidad 314 de reconstrucción (p. ej., un sumador 314) puede configurarse para añadir el bloque residual 313 reconstruido al bloque 365 de predicción para obtener un bloque reconstruido 315 en el dominio de muestra, p. ej., sumando los valores de muestra del bloque residual 313 reconstruido y los valores de muestra del bloque 365 de predicción.

Filtrado

La unidad 320 de filtro de bucle (ya sea en el bucle de codificación o después del bucle de codificación) se configura para filtrar el bloque 315 reconstruido para obtener un bloque 321 filtrado, por ejemplo, para suavizar las transiciones de píxeles, o mejorar de lo contrario la calidad de vídeo. La unidad 320 de filtro de bucle puede comprender uno o más filtros de bucle, tales como un filtro de desbloqueo, un filtro de desplazamiento adaptativo de muestra (SAO) o uno o más filtros diferentes, p. ej., un filtro de bucle adaptativo (ALF), un filtro de supresión de ruido (NSF), o cualquier combinación de los mismos. En un ejemplo, la unidad 220 de filtro de bucle puede comprender un filtro de desbloqueo, un filtro SAO y un filtro ALF. El orden del proceso de filtrado puede ser el filtro de desbloqueo, SAO y ALF. En otro ejemplo, se añade un proceso denominado mapeo de luma con escala de croma (LMCS) (es decir, el remodelador en bucle adaptativo). Este proceso se realiza antes del filtro de desbloqueo. En otro ejemplo, el proceso de filtro de desbloqueo también se puede aplicar a los bordes de subbloques internos, p. ej., los bordes de subbloques afines, los bordes de subbloques de ATMVP, los bordes de transformación de subbloques (SBT) y los bordes de intra subdivisión (ISP). Aunque la unidad 320 de filtro de bucle se muestra en la Figura 3 como un filtro de bucle, en otras configuraciones, la unidad 320 de filtro de bucle puede implementarse como un filtro de bucle posterior.

Memoria intermedia de imágenes decodificadas

Los bloques 321 de vídeo decodificados de una imagen se almacenan entonces en la memoria 330 intermedia de imágenes decodificadas, que almacena las imágenes decodificadas 331 como imágenes de referencia para la compensación de movimiento posterior para otras imágenes y/o para dar salida a una visualización, respectivamente.

El decodificador 30 está configurado para dar salida a la imagen decodificada 311, p. ej., a través de una salida 312, para su presentación a o visionado por parte de un usuario.

Predicción

La unidad 344 de interpredicción puede ser idéntica a la unidad 244 de interpredicción (en particular, a la unidad de compensación de movimiento), y la unidad 354 de intrapredicción puede ser idéntica a la unidad 254 de interpredicción en función, y toman decisiones de división o partición y realizan una predicción basándose en los parámetros de partición y/o predicción o en una información respectiva recibidos de los datos 21 de imagen codificados (p. ej., mediante un análisis y/o una decodificación, p. ej., por parte de la unidad 304 de decodificación entrópica). La unidad 360 de aplicación de modo puede configurarse para realizar la predicción (intra o interpredicción) por bloque basada en imágenes reconstruidas, bloques o muestras respectivas (filtradas o sin filtrar) para obtener el bloque 365 de predicción.

Cuando el segmento de vídeo se codifica como un segmento intracodificado (I), la unidad 354 de intrapredicción de la unidad 360 de aplicación de modo está configurada para generar un bloque 365 de predicción para un bloque de imagen del segmento de vídeo actual basada en un modo de intrapredicción señalado y en datos procedentes en bloque previamente decodificados de la imagen actual. Cuando la imagen de vídeo se codifica como un segmento intercodificado (es decir, B o P), la unidad 344 de interpredicción (p. ej., la unidad de compensación de movimiento) de la unidad 360 de aplicación de modo está configurada para producir bloques 365 de predicción para un bloque de vídeo del segmento de vídeo actual basada en los vectores de movimiento y en otros elementos de sintaxis recibidos de la unidad 304 de decodificación entrópica. Para la interpredicción, los bloques de predicción pueden producirse a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El decodificador 30 de vídeo puede construir las listas de tramas de referencia, la Lista 0 y la Lista 1, usando técnicas de construcción predeterminadas basadas en imágenes de referencia almacenadas en la DPB 330. Se puede aplicar lo mismo o algo similar para o mediante realizaciones que usen grupos de mosaicos (p. ej., grupos de

mosaicos de vídeo) y/o mosaicos (p. ej., mosaicos de vídeo) además o alternativamente a segmentos (p. ej., segmentos de vídeo), p. ej., un vídeo puede codificarse usando grupos de mosaicos y/o mosaicos I, P o B.

5 La unidad 360 de aplicación de modo se configura para determinar la información de predicción para un bloque de vídeo del segmento de vídeo actual mediante el análisis de los vectores de movimiento o información relacionada y otros elementos de sintaxis, y usa la información de predicción para producir los bloques de predicción para el bloque de vídeo actual que se está decodificando. Por ejemplo, la unidad 360 de aplicación de modo utiliza algunos de los elementos de sintaxis recibidos para determinar un modo de predicción (p. ej., intra o interpredicción) utilizado para codificar los bloques de vídeo del segmento de vídeo, un tipo de segmento de interpredicción (p. ej., un segmento B, un segmento P o un segmento GPB), información de construcción para una o más de las listas de imágenes de referencia para el segmento, vectores de movimiento para cada bloque de vídeo intercodificado del segmento, un estado de interpredicción para cada bloque de vídeo intercodificado del segmento y otra información para decodificar los bloques de vídeo en el segmento de vídeo actual. Se puede aplicar lo mismo o algo similar para o mediante realizaciones que usen grupos de mosaicos (p. ej., grupos de mosaicos de vídeo) y/o mosaicos (p. ej., mosaicos de vídeo) además o alternativamente a segmentos (p. ej., segmentos de vídeo), p. ej., un vídeo puede codificarse usando grupos de mosaicos y/o mosaicos I, P o B.

20 Las realizaciones del decodificador 30 de vídeo, tal como se muestra en la Figura 3, pueden configurarse para dividir y/o decodificar la imagen usando segmentos (también denominados segmentos de vídeo), en donde una imagen puede dividirse o decodificarse usando uno o más segmentos (típicamente sin superposición), y cada segmento puede comprender uno o más bloques (p. ej., CTU) o uno o más grupos de bloques (p. ej., mosaicos (H.265/HEVC y VVC) o ladrillos (VVC)).

25 Las realizaciones del decodificador 30 de vídeo, tal como se muestra en la Figura 3, pueden configurarse para dividir y/o decodificar la imagen usando grupos de segmentos/mosaicos (también denominados grupos de mosaicos de vídeo) y/o mosaicos (también denominados mosaicos de vídeo), en donde una imagen puede dividirse o decodificarse usando uno o más grupos de segmentos/mosaicos (típicamente sin superposición), y cada grupo de segmento/mosaico puede comprender, p. ej., uno o más bloques (p. ej., CTU) o uno o más mosaicos, en donde cada mosaico, p. ej., puede tener forma rectangular y puede comprender uno o más bloques (p. ej., CTU), p. ej., bloques completos o fraccionados.

30 Se pueden usar otras variaciones del decodificador 30 de vídeo para decodificar los datos 21 de imagen codificados. Por ejemplo, el descodificador 30 puede producir el flujo de vídeo de salida sin la unidad 320 de filtrado de bucle. Por ejemplo, un descodificador 30 basado en no transformación puede cuantizar a la inversa la señal residual directamente sin la unidad 312 de procesamiento de transformada inversa para ciertos bloques o marcos. En otra implementación, el descodificador 30 de vídeo puede tener la unidad 310 de cuantización inversa y la unidad 312 de procesamiento de transformada inversa combinadas en una sola unidad.

35 Cabe entenderse que, en el codificador 20 y en el decodificador 30, un resultado de procesamiento de una etapa actual puede procesarse aún más y luego dársele salida hacia la siguiente etapa. Por ejemplo, después del filtrado de interpolación, la obtención de vectores de movimiento o el filtrado de bucle, puede realizarse una operación adicional, tal como un corte o desplazamiento, sobre el resultado de procesamiento del filtrado de interpolación, la obtención de vectores de movimiento o el filtrado de bucle.

40 Cabe señalar que se pueden aplicar más operaciones a los vectores de movimiento obtenidos del bloque actual (incluidos, sin limitación, vectores de movimiento de punto de control del modo afín, vectores de movimiento de subbloque en los modos afín, plano y de predicción ATMVP, vectores de movimiento temporal, etc.). Por ejemplo, el valor de un vector de movimiento se restringe a un rango predefinido según su bit representativo. Si el bit representativo de un vector de movimiento es bitDepth, entonces el intervalo es $-2^{(\text{bitDepth}-1)} - 2^{(\text{bitDepth}-1)-1}$, donde el símbolo "A" quiere decir exponenciación. Por ejemplo, si bitDepth se establece igual a 16, el intervalo es -32768 - 32767; si bitDepth se establece igual a 18, el intervalo es -131072-131071. Por ejemplo, el valor del vector de movimiento derivado (p. ej., los MV de cuatro subbloques de 4x4 dentro de un bloque de 8x8) está restringido de tal modo que la diferencia máxima entre partes enteras de los cuatro MV de subbloques de 4x4 no es mayor que N píxeles, tal como no más de 1 píxel. Aquí se proporcionan dos métodos para restringir el vector de movimiento según la bitDepth.

45 La Figura 4 es un diagrama esquemático de un dispositivo 400 de codificación de vídeo según una realización de la descripción. El dispositivo 400 de codificación de vídeo es apto para implementar las realizaciones descritas tal y como se describen en la presente memoria. En una realización, el dispositivo 400 de codificación de vídeo puede ser un decodificador tal como el decodificador 30 de vídeo de la Figura 1A o un codificador tal como el codificador 20 de vídeo de la Figura 1A.

50 El dispositivo 400 de codificación de vídeo comprende unos puertos 410 de ingreso (o puertos 410 de entrada) y unas unidades receptoras (Rx) 420 para recibir datos; un procesador, unidad lógica o unidad 430 central de procesamiento (CPU) para procesar los datos; unas unidades transmisoras (Tx) 440 y unos puertos 450 de egreso (o puertos 450 de salida) para transmitir los datos; y una memoria 460 para almacenar los datos. El dispositivo 400 de codificación de vídeo también puede comprender unos componentes óptoeléctricos (OE) y unos componentes electroópticos (EO) acoplados a los puertos 410 de ingreso, a las unidades receptoras 420, a las unidades transmisoras 440 y a los puertos 450 de egreso para el ingreso o el egreso de señales ópticas o eléctricas.

65

El procesador 430 se implementa mediante hardware y software. El procesador 430 puede implementarse como uno o más chips de CPU, núcleos (p. ej., como un procesador de multinúcleo), matrices FPGA, circuitos ASIC y procesadores DSP. El procesador 430 está en comunicación con los puertos 410 de ingreso, las unidades receptoras 420, las unidades transmisoras 440, los puertos 450 de egreso y la memoria 460. El procesador 430 comprende un módulo 470 de codificación. El módulo 470 de codificación implementa las realizaciones descritas, descritas anteriormente. Por ejemplo, el módulo 470 de codificación implementa, procesa, prepara o proporciona las diversas operaciones de codificación. Por lo tanto, la inclusión del módulo 470 de codificación proporciona una mejora sustancial a la funcionalidad del dispositivo 400 de codificación de vídeo y efectúa una transformación del dispositivo 400 de codificación de vídeo a un estado diferente. Alternativamente, el módulo 470 de codificación se implementa como unas instrucciones almacenadas en la memoria 460 y ejecutadas por el procesador 430.

La memoria 460 puede comprender uno o más discos, unidades de cinta y unidades de estado sólido y puede utilizarse como un dispositivo de almacenamiento de datos de desbordamiento para almacenar programas cuando dichos programas se seleccionan para su ejecución y para almacenar instrucciones y datos que se leen durante la ejecución de programas. La memoria 460 puede ser, por ejemplo, volátil y/o no volátil y puede ser una memoria de solo lectura (ROM), una memoria de acceso aleatorio (RAM), una memoria ternaria de contenido direccionable (TCAM) y/o una memoria estática de acceso aleatorio (SRAM).

La Figura 5 es un diagrama de bloques simplificado de un aparato 500 que puede usarse como cualquiera o ambos del dispositivo 12 de origen y el dispositivo 14 de destino de la Figura 1 según una realización ejemplar.

Un procesador 502 en el aparato 500 puede ser una unidad central de procesamiento. Alternativamente, el procesador 502 puede ser cualquier otro tipo de dispositivo, o múltiples dispositivos, capaz(es) de manipular o procesar información que existe en ese momento o que se desarrollará a partir de ese momento. Aunque, tal y como se muestra, las implementaciones descritas pueden ponerse en práctica con un solo procesador, p. ej., el procesador 502, se pueden lograr ventajas de velocidad y de eficiencia usando más de un procesador.

En una implementación, una memoria 504 en el aparato 500 puede ser un dispositivo de memoria de solo lectura (ROM) o un dispositivo de memoria de acceso aleatorio (RAM). Puede usarse como memoria 504 cualquier otro tipo apropiado de dispositivo de almacenamiento. La memoria 504 puede incluir un código y unos datos 506, a los que accede el procesador 502 usando un bus 512. La memoria 504 puede incluir además un sistema operativo 508 y unos programas 510 de aplicación, incluyendo los programas 510 de aplicación al menos un programa que permite que el procesador 502 realice los métodos aquí descritos. Por ejemplo, los programas 510 de aplicación pueden incluir unas aplicaciones 1 a N, que pueden incluir además una aplicación de codificación de vídeo que realiza los métodos aquí descritos.

El aparato 500 también puede incluir uno o más dispositivos de salida, tales como una pantalla 518. En un ejemplo, la pantalla 518 puede ser un visualizador sensible al tacto que combina un visualizador con un elemento sensible al tacto que se puede hacer funcionar para detectar entradas táctiles. La pantalla 518 puede estar acoplada al procesador 502 a través del bus 512.

Aunque aquí se ha representado como un único bus, el bus 512 del aparato 500 puede estar formado por múltiples buses. Además, el almacenamiento secundario 514 puede estar directamente acoplado a los demás componentes del aparato 500 o puede accederse a él a través de una red, y puede comprender una única unidad integrada, tal como una tarjeta de memoria, o múltiples unidades, tales como múltiples tarjetas de memoria. Por lo tanto, el aparato 500 puede implementarse en una amplia variedad de configuraciones.

A continuación se describen las realizaciones de la presente invención con referencia a los dibujos adjuntos en la especificación. Debe entenderse que las realizaciones descritas en la presente memoria se utilizan simplemente para describir y explicar la presente invención, pero no pretenden limitar la presente invención.

Los escenarios ilustrativos para aplicar un filtro de desbloqueo se representan en las Figuras 6A y 6B. Como se muestra en la Figura 6A. Los bloques 601, 602, también denominados P, Q son dos bloques de codificación o bloques de transformada, el tamaño de las CU es de 16 x 4 muestras. Como se muestra en las Figuras 6A y 6B, la tecnología presentada en la presente memoria se aplica tanto a los bordes verticales como a los horizontales.

La codificación de vídeo se puede realizar basada en el espacio de color y el formato de color. Por ejemplo, el vídeo en color desempeña un papel importante en los sistemas multimedia, donde se utilizan varios espacios de color para representar el color de modo eficiente. Un espacio de color especifica el color con valores numéricos que utilizan varios componentes. Un espacio de color popular es el espacio de color RGB, donde el color se representa como una combinación de tres valores de componentes de color primarios (es decir, rojo, verde y azul). Para la compresión de vídeo en color, el espacio de color YCbCr se ha utilizado ampliamente, como se describe en A. Ford y A. Roberts, "Colour space conversions", Universidad de Westminster, Londres, Tech. Rep., agosto de 1998.

YCbCr se puede convertir fácilmente desde el espacio de color RGB mediante una transformación lineal y la redundancia entre diferentes componentes, es decir, la redundancia entre componentes, se reduce significativamente en el espacio de color YCbCr. Una ventaja de YCbCr es la retrocompatibilidad con la televisión

en blanco y negro, ya que la señal Y transmite información de luminancia. Además, la anchura de banda de crominancia se puede reducir submuestreando los componentes Cb y Cr en un formato de muestreo de croma 4:2:0 con un impacto subjetivo significativamente menor que el submuestreo en el espacio de color RGB. Debido a estas ventajas, YCbCr ha sido el principal espacio de color en la compresión de vídeo. También hay otros espacios de color, tales como YCoCg, que se utilizan en la compresión de vídeo. En esta descripción, independientemente del espacio de color real utilizado, la luma (o L o Y) y las dos cromas (Cb y Cr) se utilizan para representar los tres componentes de color en el esquema de compresión de vídeo. Por ejemplo, cuando la estructura de muestreo en formato de croma es un muestreo 4:2:0, cada una de las dos matrices de croma tiene la mitad de la altura y la mitad de la anchura de la matriz de luma. En la Figura 7A se muestra un ejemplo de las ubicaciones relativas nominales verticales y horizontales de las muestras de luma y croma en las imágenes. La Figura 7B ilustra un ejemplo de muestreo 4:2:0. La Figura 7B ilustra un ejemplo de un bloque de luma y un bloque de croma coubicados. Si el formato de vídeo es YUV4:2:0, entonces hay un bloque de luma de 16x16 y dos bloques de croma de 8x8.

Específicamente, un bloque de codificación o un bloque de transformada contiene un bloque de luma y dos bloques de croma. Como se muestra, el bloque de luma contiene cuatro veces más muestras que el bloque de croma. Específicamente, el bloque de croma contiene un número N de muestras por un número N de muestras, mientras que el bloque de luma contiene un número 2N de muestras por un número 2N de muestras. Por lo tanto, el bloque de luma tiene una resolución cuatro veces superior a la del bloque de croma. Por ejemplo, cuando se usa el formato YUV4:2:0, las muestras de luma se pueden reducir en un factor de cuatro (p. ej., dos de anchura y dos de altura). YUV es un sistema de codificación de color que emplea un espacio de color en términos de componentes de luma Y y dos componentes de crominancia U y V.

Un ejemplo de un escenario para aplicar un filtro de desbloqueo para dos bloques de luma se representa en la Figura 8. Hay un borde 803 de bloque de luma entre un primer bloque 801 de luma de un primer bloque (601, 601') de imagen y un segundo bloque 802 de luma de un segundo bloque (602, 602') de imagen.

Los escenarios en los que se aplica el filtro de desbloqueo para dos bloques de croma se representan en las Figuras 9A-9H. La Figura 9A ilustra un bloque 901 de croma P de un primer bloque (601, 601') de imagen, un bloque 902 de croma Q de un segundo bloque (602, 602') de imagen y un borde 903 de bloque de croma al que se aplica el filtro de desbloqueo. Como se muestra en la Figura 9A, un borde 903 de bloque de croma vertical entre los bloques 901, 902 de croma se filtra según la realización de la presente invención. Cada una de las Figuras 9B-9H muestra un ejemplo del bloque de croma para el bloque de croma P y el bloque de croma Q. Por ejemplo, en la Figura 9B, el borde 913 vertical del componente Cb entre los componentes Cb 911, 912 se filtra según la realización de la presente invención. En la Figura 9C, un borde 923 vertical del componente Cr entre los componentes Cr 921, 922 se filtra según la realización de la presente invención.

En el diseño para desbloquear la VVC anterior, siempre que un bloque de croma dado (Cb o Cr) utiliza una herramienta de codificación conjunta de Cb-Cr (JCCR), el QP correspondiente utilizado en el desbloqueo de croma se deriva de la tabla de mapeo de ChromaQP diseñada para los componentes individuales Cb y Cr. Para los bloques codificados de Cb-Cr conjuntos o los componentes Cb-Cr conjuntos, el uso de ChromaQPTable diseñada para los componentes Cb y Cr puede resultar en una derivación incorrecta del QP de croma y, por lo tanto, las decisiones de desbloqueo y/o el proceso de desbloqueo (que depende del valor del QP de croma) se ven afectados. Esto da como resultado decisiones de desbloqueo y/o un proceso de desbloqueo inexactos y, por lo tanto, puede conducir a artefactos de bloque que permanecen en la imagen, lo que afecta a la calidad visual general de la imagen codificada.

En el diseño anterior de desbloqueo de la codificación de vídeo versátil (VVC), el QP de croma se obtiene del siguiente modo:

Las variables Q_{pQ} y Q_{pP} se establecen iguales a los valores Q_{pY} de las unidades de codificación que incluyen los bloques de codificación que contienen la muestra $q_{0,0}$ y $p_{0,0}$, respectivamente. Puede entenderse que la variable Q_{pQ} representa el valor Q_{pY} para un bloque de luma Q correspondiente, y la variable Q_{pP} representa el valor de Q_{pY} para un bloque de luma P correspondiente.

La variable Q_{pC} se deriva del siguiente modo:

$$q_{p_i} = \text{Clip3}(0, 63, ((Q_{pQ} + Q_{pP} + 1) \gg 1) + c_{QpPicOffset}) \quad (8-1132)$$

$$Q_{pC} = \text{ChromaQpTable}[cIdx - 1][q_{p_i}] \quad (8-1133)$$

Por lo tanto, el valor de QP de croma utilizado en el filtro de desbloqueo de croma se deriva básicamente promediando los valores de QP de luma y, a continuación, se usa ChromaQPTable para mapear el QP de luma promediado al QP de croma.

El problema del esquema de desbloqueo anterior es que, para los bloques codificados de Cb-Cr conjuntos (JCCR) o los componentes Cb-Cr conjuntos, el mapeo de QP de croma también se realiza utilizando ChromaQPTable, que se diseña específicamente para los componentes Cb y Cr.

Las realizaciones de la presente invención tienen como objetivo mejorar el filtrado de desbloqueo convencional. La descripción presentada en la presente memoria implica un aparato de filtro de desbloqueo, un codificador, un

decodificador y los métodos correspondientes que pueden realizar el desbloqueo de un borde de bloque (903, 913, 923, 933, 943, 953, 963, 973) de croma entre un primer bloque (901, 911, 921, 931, 941, 951, 961, 971) de croma de un primer bloque (601, 601') de imagen y un segundo bloque (902, 912, 922, 932, 942, 952, 962, 972) de croma de un segundo bloque (602, 602') de imagen de forma precisa. Además, el filtro de desbloqueo debe ser eficiente y preciso.

5 Realizaciones de la implementación técnica de la presente invención

Según una realización de la presente invención, siempre que un bloque de croma dado (Cb o Cr) utilice herramientas de codificación de croma conjunta (JCCR), se determina un parámetro de cuantificación de croma Qpc basado en el parámetro de cuantificación de luma promediado utilizando una regla de conversión que es diferente en comparación con el componente de color Cb y Cr.

Según una realización de la presente invención, siempre que un bloque de croma determinado (Cb o Cr) utilice herramientas de codificación de croma conjunta (JCCR), el QP correspondiente que se usa en el desbloqueo de croma debe derivarse utilizando ChromaQPTable, que se diseña para bloques codificados de Cb-Cr conjuntos o componentes de Cb-Cr conjuntos (abreviatura en bloque de JCCR).

La tabla de mapeo de QP de croma diseñada específicamente para bloques de JCCR cuyo índice se representa mediante un valor entero. El valor del índice no es el mismo que el valor del índice de Cb y Cr y puede tener valores de 3, 4, 5...

En un ejemplo, como se ilustra en la Figura 13, cada componente de croma (Cb, Cr o Cb-Cr conjuntos) tiene su propia tabla de mapeo de QP de croma. En otro ejemplo, como se ilustra en las Figuras 12A y 12B, una sola tabla con tres entradas diferentes para Cb, Cr y Cb-Cr conjuntos.

En un aspecto, la presente invención consiste en derivar correctamente el QP de croma que se usa cuando al menos uno de los bloques de croma contiguos entre sí son bloques codificados de Cb-Cr conjuntos (JCCR) o cuando el al menos uno de los bloques de croma se codifica usando el modo de Cb-Cr conjuntos (JCCR).

En otro aspecto, la presente invención consiste en llevar a cabo las decisiones de desbloqueo y/o el proceso de filtrado de desbloqueo que dependen indirectamente del valor de QP de croma que se deriva correctamente y, por lo tanto, los artefactos de bloque pueden reducirse y, a su vez, mejorar la calidad subjetiva general.

Primera realización de la presente invención

Según un primer aspecto no cubierto por la invención reivindicada, este ejemplo se refiere a un método de desbloqueo, para desbloquear un borde de bloque de croma entre un primer bloque de croma de un primer bloque de imagen y un segundo bloque de croma de un segundo bloque de imagen, en una codificación de imagen y/o una decodificación de imagen,

en donde el método de desbloqueo comprende:

– Realizar un proceso de decisión para el borde del bloque de croma, en donde el proceso de decisión comprende:

en el caso de que al menos uno del primer bloque de croma y el segundo bloque de croma sea un bloque codificado de Cb-Cr conjuntos (JCCR) (o al menos uno del primer bloque de croma y el segundo bloque de croma utilice herramientas conjuntas de residuos de Cb-Cr (JCCR) o al menos uno del primer bloque de croma y el segundo bloque de croma se codifique usando el modo de Cb-Cr conjuntos (JCCR)),

determinar un parámetro de cuantificación de luma qPi promediado basado en un primer QP de luma (tal como QpYP) de un primer bloque de luma del primer bloque de imagen y un segundo QP de luma (tal como QpYO) de un segundo bloque de luma del segundo bloque de imagen; y

determinar un parámetro de cuantificación de croma Qpc basado en el parámetro de cuantificación de luma qPi promediado usando una tabla de mapeo de Qp de croma (tal como ChromaQPTable) cuyo índice es un primer valor de índice o usando una tabla de mapeo de Qp de croma que comprende entradas de información cuyo índice es un primer valor de índice, en donde el primer valor de índice indica que al menos uno del primer bloque de croma y el segundo bloque de croma es un bloque codificado de Cb-Cr conjuntos (JCCR) o el primer valor de índice corresponde al caso en el que el al menos uno del primer bloque de croma y el segundo bloque de croma usa herramientas de JCCR;

– realizar 1603 un proceso de filtrado para el borde del bloque de croma basado en un resultado de decisión del proceso de decisión.

el resultado de la decisión indica la determinación de si el borde (403, 504, 903) del bloque debe filtrarse y/o si se debe realizar un filtrado prolongado.

65

En una posible forma de implementación del método según el primer aspecto como tal, en donde el primer valor de índice corresponde a una tabla de mapeo de Q_p de croma, o el primer valor de índice corresponde a entradas de información de una tabla de mapeo de Q_p de croma.

5 En una posible forma de implementación del método según cualquier implementación anterior del primer aspecto o del primer aspecto como tal, en donde el primer valor de índice es diferente del valor de un índice de componente (Cb, Cr) (cldx), o el primer valor de índice es diferente del valor de un índice de componente (Cb, Cr) (cldx) menos 1.

10 En una posible forma de implementación del método según cualquier implementación anterior del primer aspecto o del primer aspecto como tal, en donde el primer valor de índice es 2, o el primer valor de índice es un componente (componente Cb-Cr conjunto), siendo el valor del índice (cldx) 3.

15 En una posible forma de implementación del método según cualquier implementación anterior del primer aspecto o del primer aspecto como tal, en donde el parámetro de cuantificación de croma Q_{pc} se usa para el proceso de decisión para el borde de bloque de croma, o el parámetro de cuantificación de croma Q_{pc} se usa directa o indirectamente para determinar si el borde del bloque de croma se debe filtrar y/o si se debe realizar un filtrado prolongado.

20 En una posible forma de implementación del método según cualquier implementación anterior del primer aspecto o del primer aspecto como tal, en donde

el valor de una primera variable de umbral (tal como β') se determina basado en el parámetro de cuantificación de croma Q_{pc} o un primer Q_{pc} recortado usando una tabla de mapeo; y el valor de una primera variable (tal como β) se deriva basada en el valor de la primera variable de umbral (tal como β');

25 en donde el valor de la primera variable (tal como β) se usa para el proceso de decisión para el borde de bloque de croma o el valor de la primera variable (tal como β) se usa para determinar si el borde de bloque de croma se debe filtrar y/o si se debe realizar un filtrado prolongado.

30 En una posible forma de implementación del método según cualquier implementación anterior del primer aspecto o del primer aspecto como tal, en donde

el valor de una segunda variable de umbral (tal como tc') se determina basado en el parámetro de cuantificación de croma Q_{pc} o un segundo Q_{pc} recortado usando una tabla de mapeo;

35 el valor de una segunda variable (tal como tc) se deriva basado en el valor de la segunda variable de umbral (tal como tc');

40 en donde el valor de la segunda variable (tal como tc) se usa para el proceso de decisión para el borde de bloque de croma (o el valor de la segunda variable [tal como tc] se usa para determinar si el borde de bloque de croma se debe filtrar y/o si se debe realizar un filtrado prolongado) y el proceso de filtrado para el borde de bloque de croma.

En una posible forma de implementación del método según cualquier implementación anterior del primer aspecto o del primer aspecto como tal, en donde la tabla de mapeo comprende el mapeo entre una pluralidad de las primeras variables de umbral, una pluralidad de las segundas variables de umbral y una pluralidad de parámetros de cuantificación.

45 En una posible forma de implementación del método según cualquier implementación anterior del primer aspecto o del primer aspecto como tal, en donde el primer bloque de croma es un primer componente de croma (tal como el componente Cb) del primer bloque de imagen y el segundo bloque de croma es un primer componente de croma (tal como el componente Cb) del segundo bloque de imagen, y/o en donde el primer bloque de croma es un segundo componente de croma (tal como el componente Cr) del primer bloque de imagen y el segundo bloque de croma es un segundo componente de croma (tal como el componente Cr) del segundo bloque de imagen.

50 En una posible forma de implementación del método según cualquier implementación anterior del primer aspecto o del primer aspecto como tal, en donde al menos uno del primer componente de croma (tal como el componente Cb) del primer bloque de imagen y el primer componente de croma (tal como el componente Cb) del segundo bloque de imagen son los componentes Cb-Cr conjuntos o se codifican usando el modo de Cb-Cr conjuntos (JCCR), y/o

55 en donde al menos uno del segundo componente de croma (tal como el componente Cr) del primer bloque de imagen y el segundo componente de croma (tal como el componente Cr) del segundo bloque de imagen son los componentes Cb-Cr conjuntos o se codifican usando el modo de Cb-Cr conjuntos (JCCR), o

60 en donde al menos uno del primer y segundo bloque de croma es un bloque codificado de Cb-Cr conjuntos (JCCR). En una posible forma de implementación del método según cualquier implementación anterior del primer aspecto o del primer aspecto como tal, en donde el primer bloque de imagen y el segundo bloque de imagen son bloques de transformada;

65 o

el primer bloque de imagen y el segundo bloque de imagen son bloques de codificación.

5 En una posible forma de implementación del método según cualquier implementación anterior del primer aspecto o del primer aspecto como tal, en donde se determina si al menos uno del primer bloque de croma y el segundo bloque de croma usa herramientas de codificación conjunta de residuos de crominancia (JCCR) basadas en un indicador de nivel de TU de unidad de transformada (tal como `tu_joint_cbcr_residual_flag`).

10 En una posible forma de implementación del método según cualquier implementación anterior del primer aspecto o del primer aspecto como tal, en donde cuando el indicador de nivel de TU (tal como `tu_joint_cbcr_residual_flag`) para el primer bloque de imagen es verdadero, el primer bloque de croma usa herramientas de codificación conjunta de residuos de crominancia (JCCR) (o el primer bloque de croma se codifica usando el modo de Cb-Cr conjuntos (JCCR), o el primer bloque de croma es un bloque codificado de Cb-Cr conjuntos (JCCR) o un componente Cb-Cr conjunto); o cuando el indicador de nivel de TU (tal como `tu_joint_cbcr_residual_flag`) para el segundo bloque de imagen es verdadero, el segundo bloque de croma usa herramientas de codificación conjunta de residuos de crominancia (JCCR) (o el segundo bloque de croma se codifica usando el modo de Cb-Cr conjuntos (JCCR) o el segundo bloque de croma es un bloque codificado de Cb-Cr conjuntos (JCCR) o el segundo bloque de croma es un bloque codificado de Cb-Cr conjuntos (JCCR) o un componente Cb-Cr conjunto).

20 En una posible forma de implementación del método según cualquier implementación anterior del primer aspecto o del primer aspecto como tal, en donde la tabla de mapeo de Qp de croma se diseña específicamente para el bloque codificado de Cb-Cr conjuntos (JCCR), o la tabla de mapeo de Qp de croma se diseña para el primer componente de croma (tal como el componente Cb), el segundo componente de croma (tal como el componente Cr) y el componente Cb-Cr conjunto.

25 En una posible forma de implementación del método según cualquier implementación anterior del primer aspecto o del primer aspecto como tal, en donde

30 la tabla de mapeo de Qp de croma (tal como ChromaQPTable) comprende la correspondencia entre una pluralidad de parámetros de cuantificación de croma Q_{p_c} y una pluralidad de parámetros de cuantificación de luma q_{Pi} , y la pluralidad de parámetros de cuantificación de croma Q_{p_c} se asocia con el primer valor de índice; o

la tabla de mapeo de Qp de croma (tal como ChromaQPTable) comprende:

35 un primer conjunto de parámetros de cuantificación de croma $Q_{p_{JCCR}}$, un segundo conjunto de parámetros de cuantificación de croma $Q_{p_{cb}}$ y un tercer conjunto de parámetros de cuantificación de croma $Q_{p_{cr}}$, cada uno del primer conjunto, segundo conjunto y tercer conjunto corresponde a una pluralidad de parámetros de cuantificación de luma q_{Pi} , en donde el primer conjunto de parámetros de cuantificación de croma $Q_{p_{JCCR}}$ se asocia con el primer valor de índice; el segundo conjunto de parámetros de cuantificación de croma $Q_{p_{cb}}$ se asocia con un segundo valor de índice; y el tercer conjunto de parámetros de cuantificación de croma $Q_{p_{cr}}$ se asocia con un tercer valor de índice.

40 En una posible forma de implementación del método según cualquier implementación anterior del primer aspecto o del primer aspecto como tal, en donde el primer valor de índice (tal como 3) indica que al menos uno del primer bloque de croma y el segundo bloque (402, 502) de croma es un bloque codificado de Cb-Cr conjuntos (JCCR).

45 En una posible forma de implementación del método según cualquier implementación anterior del primer aspecto o del primer aspecto como tal, en donde el primer valor de índice o el índice de componentes Cb-Cr conjunto (`cldx`) es un valor entero tal como 3, 4, 5,...

50 En una posible forma de implementación del método según cualquier implementación anterior del primer aspecto o del primer aspecto como tal, en donde

55 en el caso de que el primer bloque de croma sea un primer componente de croma (tal como el componente Cb) del primer bloque de imagen y el segundo bloque de croma sea un primer componente de croma (tal como el componente Cb) del segundo bloque de imagen, el índice de componente de color (`cldx`) tiene un segundo valor que es 1; o

en el caso de que el primer bloque de croma sea un segundo componente de croma (tal como el componente Cr) del primer bloque de imagen y el segundo bloque de croma sea un segundo componente de croma (tal como el componente Cr) del segundo bloque de imagen, el índice de componente de color (`cldx`) tiene un tercer valor que es 2; o

60 en el caso de que el primer bloque de croma sea un primer componente de croma (tal como el componente Cb) del primer bloque de imagen y el segundo bloque de croma sea un primer componente de croma (tal como el componente Cb) del segundo bloque de imagen, y al menos uno de los primeros componentes de croma sea el componente Cb-Cr conjunto (JCCR), el índice de componente de color (`cldx`) tiene el primer valor de 3; o

65 en el caso de que el primer bloque de croma sea un segundo componente de croma (tal como el componente Cr) del primer bloque de imagen y el segundo bloque de croma sea un segundo componente de croma (tal como el

componente Cr) del segundo bloque de imagen, y al menos uno de los segundos componentes de croma sea el componente de Cb-Cr conjuntos (JCCR), el índice de componente de color (cldx) tiene el primer valor de 3; o

5 en el caso de que al menos uno del primer y segundo bloque de croma sea un bloque codificado de Cb-Cr conjuntos (JCCR), el índice de componentes de color (cldx) tiene un primer valor de 3.

En una posible forma de implementación del método según cualquier implementación anterior del primer aspecto o del primer aspecto como tal, en donde la tabla de mapeo de Qp de croma (tal como ChromaQPTable) comprende:

10 un primer conjunto de parámetros de cuantificación de croma $Q_{p_{JCCR}}$, un segundo conjunto de parámetros de cuantificación de croma $Q_{p_{cb}}$ y un tercer conjunto de parámetros de cuantificación de croma $Q_{p_{cr}}$, cada uno del primer conjunto, segundo conjunto y tercer conjunto corresponde a un cuarto conjunto de parámetros de cuantificación de luma q_{Pi} , en donde el primer conjunto de parámetros de cuantificación de croma $Q_{p_{JCCR}}$ se asocia con el índice de componente de color (cldx) que tiene el primer valor; el segundo conjunto de parámetros de cuantificación de croma $Q_{p_{cb}}$ se asocia con el índice de componente de color (cldx) que tiene el segundo valor; y el tercer conjunto de parámetros de cuantificación de croma $Q_{p_{cr}}$ se asocia con el índice de componente de color (cldx) que tiene el tercer valor.

20 En una posible forma de implementación del método según cualquier implementación anterior del primer aspecto o del primer aspecto como tal, en donde la tabla de mapeo de Qp de croma (tal como ChromaQPTable) comprende la correspondencia entre una pluralidad de parámetros de cuantificación de croma Q_{p_c} y una pluralidad de parámetros de cuantificación de luma Q_{p_i} , y la pluralidad de parámetros de cuantificación de croma Q_{p_c} se asocia con el primer valor del índice de componente (componente Cb-Cr conjunto) (cldx).

25 En una posible forma de implementación del método según cualquier implementación anterior del primer aspecto o del primer aspecto como tal, en donde el primer bloque (401, 501) de croma tiene un tamaño de bloque que es $M \times N$ o $N \times M$, y M y N representan la anchura y la altura del primer bloque de croma respectivamente, o N y M representan la anchura y la altura del primer bloque de croma respectivamente;

30 en donde el segundo bloque de croma (402, 502) tiene un tamaño de bloque que es $L \times T$ o $T \times L$, y L y T representan la anchura y la altura del primer bloque de croma respectivamente, o T y L representan la anchura y la altura del primer bloque de croma, respectivamente,

35 en donde N o T es un número entero par 2^n mayor o igual a un umbral (tal como 4, 8, 16, 32...) y n es un número entero positivo.

En una posible forma de implementación del método según cualquier implementación anterior del primer aspecto o del primer aspecto como tal,

40 – si el borde (903) de bloque de croma es un borde (903) de bloque de croma horizontal, la dirección a lo largo de la altura N del primer bloque de croma es perpendicular al borde (903) de bloque de croma, y la dirección a lo largo de la altura T del segundo bloque de croma es perpendicular al borde (903) de bloque de croma, en donde una o ambas alturas del primer y segundo bloque de croma son un entero par 2^n mayor que 4 (o mayor o igual a 8); o

45 – si el borde (903) de bloque de croma es un borde (403, 504) de bloque de croma vertical, la dirección a lo largo de la anchura N del primer bloque de croma es perpendicular al borde (903) del bloque, y la dirección a lo largo de la anchura T del segundo bloque de croma es perpendicular al borde (903) del bloque, en donde una o ambas anchuras del primer y segundo bloque de croma son un número entero par 2^n mayor que 4 (o mayor o igual a 8).

50 En una posible forma de implementación del método según cualquier implementación anterior del primer aspecto o del primer aspecto como tal, en donde el segundo bloque de imagen es un bloque actual y el primer bloque de imagen es un bloque vecino del bloque actual.

55 **Solución 1:** se realiza una comprobación si el bloque P o el bloque Q o ambos bloques P y Q utilizan la herramienta Cb-Cr conjunta (JCCR). Si ninguno de los bloques P y Q usa JCCR, entonces el mapeo de QP de croma se realiza de la misma forma en que se hace actualmente para los bloques de Cb y Cr (consultar las ecuaciones 8-1132 y 8-1133). Pero si uno de los bloques usa la herramienta JCCR, entonces el valor promedio de QP de luma de los dos bloques P y Q se deriva de la misma forma en que se diseña actualmente en la VVC 6.0 (consultar la ecuación 8-1132). Además, cuando se mapea el Qp de luma en el QP de croma, a modo de ejemplo, se usa la tabla de mapeo de QP de croma que se diseña para el bloque codificado de Cb-Cr conjuntos (JCCR) o el componente Cb-Cr conjunto, de modo alternativo, se puede usar la tabla de mapeo de QP de croma que se diseña para el componente Cb-Cr conjunto y los componentes de croma (Cb, Cr), como se muestra en una ecuación 8-1133';

60 Por lo tanto, la ecuación 8-1132 sigue siendo la misma, pero la ecuación 8-1133', que es diferente de la ecuación 8-1133, se ilustra del siguiente modo:

65
$$q_{Pi} = \text{Clip3}(0, 63, ((Q_{pQ} + Q_{pP} + 1) \gg 1) + c_{QPpicOffset}) \quad (8-1132)$$

$$Q_{pc} = \text{ChromaQpTable}[2][q_{Pi}] \quad (8-1133)$$

5 Los cambios en el texto de la especificación para la solución 1 se proporcionan con respecto a la sección 8.8.3.6.3 del documento JVET-O2001 (versión - vE), del siguiente modo.

8.8.3.6.3 Proceso de decisión para bordes en bloque de croma

10 Las variables Q_{pQ} y Q_{pP} se establecen iguales a los valores Q_{pY} de las unidades de codificación que incluyen los bloques de codificación que contienen la muestra $q_{0,0}$ y $p_{0,0}$, respectivamente.

La variable Q_{pc} se deriva del siguiente modo:

$$15 \quad q_{Pi} = \text{Clip3}(0, 63, ((Q_{pQ} + Q_{pP} + 1) \gg 1) + c_{QpPicOffset}) \quad (8-1132)$$

– Si la muestra $p_{0,0}$ o $q_{0,0}$ está en una unidad de transformada con $tu_joint_cbcr_residual_flag$ igual a 1, entonces

- $Q_{pc} = \text{ChromaQpTable}[2][q_{Pi}] \quad (8-1133)$

20 – De lo contrario

- $Q_{pc} = \text{ChromaQpTable}[cldx - 1][q_{Pi}] \quad (8-1133)$

25 Cabe señalar que:

$cldx = 0$ si los bloques que contienen la muestra $q_{0,0}$ y $p_{0,0}$ son bloques de luma o componentes de luma,

$cldx = 1$ si los bloques que contienen la muestra $q_{0,0}$ y $p_{0,0}$ son los primeros componentes de croma (Cb),

30 $cldx = 2$ si los bloques que contienen la muestra $q_{0,0}$ y $p_{0,0}$ son los segundos componentes de croma (Cr), y

$cldx = 3$ si uno o ambos de los bloques que contienen las muestras $q_{0,0}$ y $p_{0,0}$ son bloques codificados Cb-Cr conjuntos (JCCR) o componentes Cb-Cr conjuntos.

35 La etapa de determinación de si el bloque P o el bloque Q o ambos bloques P y Q utilizan la herramienta Cb-Cr conjuntos (JCCR) puede incluir:

40 si la muestra $p_{0,0}$ está en una unidad de transformada con $tu_joint_cbcr_residual_flag$ igual a 1, entonces se determina que el bloque P usa la herramienta Cb-Cr conjuntos (JCCR); y/o

si la muestra $q_{0,0}$ está en una unidad de transformada con $tu_joint_cbcr_residual_flag$ igual a 1, se determina que el bloque Q utiliza la herramienta Cb-Cr conjuntos (JCCR).

45 Los efectos beneficiosos y las ventajas de la solución 1 según una realización de la presente invención incluyen que la indexación de la tabla de mapeo ChromaQpTable se modifique basada en la comprobación de que el bloque actual usa JCCR o no sin cambios significativos en el mapeo de conversión.

Segunda forma de realización de la presente invención

50 Según un segundo aspecto, la invención se refiere a un método de desbloqueo para desbloquear un borde 903, 913, 923 de bloque de croma entre un primer bloque 901, 911, 921 de croma de un primer bloque 601, 601' de imagen y un segundo bloque 902, 912, 922 de croma de un segundo bloque 602, 602' de imagen, en una codificación de imagen y/o una decodificación de imagen,

55 en donde el método de desbloqueo comprende:

– realizar un proceso de decisión para el borde de bloque de croma, en donde el proceso de decisión comprende:

60 ▪ determinar un parámetro de cuantificación de croma Q_{pCb} para el primer bloque 901, 911, 921 de croma basado en un primer QP de luma (tal como Q_{pYP}) de un primer bloque 801, 802 de luma del primer bloque 901, 911, 921 de imagen utilizando una tabla de mapeo de Qp de croma (tal como ChromaQpTable [0], ChromaQpTable [1], ChromaQpTable [2] de la tabla 1200, 1200', 1301, 1302, 1303) o entradas de información de la tabla de mapeo de Qp de croma, que se indica mediante un índice (tal como índice de componente, $cldx$ o índice de fila), y el índice que tiene un primer valor de índice (tal como 3) cuando el primer bloque de croma es un bloque codificado de Cb-Cr conjuntos (JCCR), o el índice ($cldx$) que tiene un segundo valor de índice (tal como 1 o 2) cuando el primer bloque de croma es el componente de croma del primer bloque de imagen;

65

5 ▪ determinar un parámetro de cuantificación de croma Qp_{Cq} para el segundo bloque 902, 912, 922 de croma basado en un segundo QP de luma (tal como Qp_{YQ}) de un segundo bloque (401, 501) de luma del segundo bloque de imagen usando una tabla de mapeo de Qp de croma (tal como ChromaQPTable) o entradas de información de la tabla de mapeo de Qp de croma que se indica mediante un índice (tal como un índice de componente, $cldx$) y el índice ($cldx$) que tiene un primer valor (tal como 3) cuando el segundo bloque de croma es un bloque codificado de Cb-Cr conjuntos (JCCR), o el índice ($cldx$) que tiene un segundo valor (tal como 1 o 2) cuando el segundo bloque de croma es el componente de croma del segundo bloque de imagen; y

10 ▪ determinar un parámetro de cuantificación de croma Qpc promediado y redondeado basado en el parámetro de cuantificación de croma Qp_{Cp} para el primer bloque de croma y el parámetro de cuantificación de croma Qp_{Cq} para el segundo bloque de croma;

15 – realizar un proceso de filtrado para el borde de bloque de croma basado en un resultado de decisión del proceso de decisión.

Puede entenderse que el resultado de la decisión indica que el borde del bloque debe filtrarse o no, y/o que debe realizarse o no un filtrado prolongado.

20 En una posible forma de implementación del método según el segundo aspecto como tal, en donde diferentes valores de índice corresponden a diferentes entradas de información de una tabla de mapeo de QP de croma, o diferentes valores de índice corresponden a diferentes tablas de mapeo de QP de croma.

25 En una posible forma de implementación del método según cualquier implementación anterior del segundo aspecto o del segundo aspecto como tal, en donde la etapa de determinación de un parámetro de cuantificación de croma Qpc promediado y redondeado basado en el parámetro de cuantificación de croma Qp_{Cp} para el primer bloque de croma y el parámetro de cuantificación de croma Qp_{Cq} para el segundo bloque de croma, comprende:

30 determinar un parámetro de cuantificación del primer componente de croma (tal como el componente Cb) $QpCb$ promediado basado en

– el parámetro de cuantificación del primer componente de croma (tal como el componente Cb) $QpCb_p$ para el primer componente de croma (tal como el componente Cb) del primer bloque de imagen y

35 – el parámetro de cuantificación del primer componente de croma (tal como el componente Cb) $QpCb_q$ para el primer componente de croma (tal como el componente Cb) del segundo bloque de imagen.

40 En una posible forma de implementación del método según cualquier implementación anterior del segundo aspecto o del segundo aspecto como tal, en donde la etapa de determinación de un parámetro de cuantificación de croma Qpc promediado y redondeado basado en el parámetro de cuantificación de croma Qp_{Cp} para el primer bloque de croma y el parámetro de cuantificación de croma Qp_{Cq} para el segundo bloque de croma, comprende:

45 determinar un parámetro de cuantificación del segundo componente de croma (tal como el componente Cr) $QpCr$ promediado basado en

– el parámetro de cuantificación del segundo componente de croma (tal como el componente de Cr) $QpCr_p$ para el segundo componente de croma (tal como el componente de Cr) del primer bloque de imagen y

50 – el parámetro de cuantificación del segundo componente de croma (tal como el componente de Cr) $QpCr_q$ para el segundo componente de croma (tal como el componente de Cr) del segundo bloque de imagen.

55 En una posible forma de implementación del método según cualquier implementación anterior del segundo aspecto o del segundo aspecto como tal, en donde el parámetro de cuantificación de croma Qp_c promediado y redondeado se usa para el proceso de decisión para el borde de bloque de croma, o el parámetro de cuantificación de croma Qp_c promediado y redondeado se usa directa o indirectamente para determinar si el borde del bloque de croma debe filtrarse y/o si debe realizarse un filtrado prolongado.

60 En una posible forma de implementación del método según cualquier implementación anterior del segundo aspecto o del segundo aspecto como tal, en donde

el valor de una primera variable de umbral (tal como β') se determina basada en el parámetro de cuantificación de croma Qpc promediado y redondeado o un primer Qpc recortado utilizando una tabla de mapeo; y el valor de una primera variable (tal como β) se deriva basada en el valor de la primera variable de umbral (tal como β');

65

en donde el valor de la primera variable (tal como β) se usa para el proceso de decisión para el borde del bloque de croma o se usa para determinar si el borde del bloque se debe filtrar y/o si se debe realizar un filtrado prolongado.

Según la invención reivindicada,

5 el valor de una segunda variable de umbral (tal como t_c') se determina basada en el parámetro de cuantificación de croma Q_{pc} promediado y redondeado;

10 el valor de una segunda variable (tal como t_c) se deriva basado en el valor de la segunda variable de umbral (tal como t_c');
 en donde el valor de la segunda variable (tal como t_c) se usa para el proceso de decisión para el borde del bloque de croma (o para determinar si el borde del bloque se debe filtrar y/o si se debe realizar un filtrado prolongado) y el proceso de filtrado para el borde del bloque de croma.

15 En una posible forma de implementación del método según cualquier implementación anterior del segundo aspecto o del segundo aspecto como tal, en donde la tabla de mapeo comprende el mapeo entre una pluralidad de las primeras variables de umbral, una pluralidad de las segundas variables de umbral y una pluralidad de parámetros de cuantificación.

20 En una posible forma de implementación del método según cualquier implementación anterior del segundo aspecto o del segundo aspecto como tal, en donde el primer bloque de croma es un primer componente de croma (tal como el componente Cb) del primer bloque de imagen y el segundo bloque de croma es un primer componente de croma (tal como el componente Cb) del segundo bloque de imagen, o

25 en donde el primer bloque de croma es un segundo componente de croma (tal como el componente Cr) del primer bloque de imagen y el segundo bloque de croma es un segundo componente de croma (tal como el componente Cr) del segundo bloque de imagen.

30 En una posible forma de implementación del procedimiento según cualquier implementación anterior del segundo aspecto o del segundo aspecto como tal, en donde el primer bloque de imagen y el segundo bloque de imagen son unidades de transformada o bloques de transformada.

35 En una posible forma de implementación del método según cualquier implementación anterior del segundo aspecto o del segundo aspecto como tal, en donde se determina si al menos uno del primer bloque de croma y el segundo bloque (402, 502) de croma se codifica usando el modo de Cb-Cr conjuntos (JCCR) o si al menos uno del primer bloque de croma y el segundo bloque (402, 502) de croma es un bloque codificado de Cb-Cr conjuntos (JCCR) basado en un indicador de nivel de TU de unidad de transformada (tal como $tu_joint_cbcr_residual_flag$).

40 En una posible forma de implementación del método según cualquier implementación anterior del segundo aspecto o del segundo aspecto como tal, en donde cuando el indicador de nivel de TU (tal como $tu_joint_cbcr_residual_flag$) para el primer bloque de imagen es verdadero, el primer bloque de croma usa herramientas de codificación conjunta de residuos de crominancia (JCCR) o el primer bloque de croma se codifica usando el modo de Cb-Cr conjuntos (JCCR); o

45 cuando el indicador de nivel de TU (tal como $tu_joint_cbcr_residual_flag$) para el segundo bloque de imagen es verdadero, el segundo bloque de croma usa herramientas de codificación conjunta de residuos de crominancia (JCCR) o el segundo bloque de croma se codifica usando el modo de Cb-Cr conjuntos (JCCR).

50 En una posible forma de implementación del método según cualquier implementación anterior del segundo aspecto o del segundo aspecto como tal, en donde la tabla de mapeo de Q_p de croma (tal como ChromaQPTable) comprende lo correspondiente entre una pluralidad de parámetros de cuantificación de croma Q_{p_c} y una pluralidad de parámetros de cuantificación de luma $Q_{p_{luma}}$, y la pluralidad de parámetros de cuantificación de croma Q_{p_c} se asocia con el primer valor de índice o el segundo valor de índice.

55 En una posible forma de implementación del método según cualquier implementación anterior del segundo aspecto o del segundo aspecto como tal, el primer bloque de croma tiene un tamaño de bloque que es $M \times N$ o $N \times M$, y M y N representan la anchura y la altura del primer bloque de croma respectivamente, o N y M representan la anchura y la altura del primer bloque de croma respectivamente; en donde el segundo bloque de croma tiene un tamaño de bloque que es $L \times T$ o $T \times L$, y L y T representan la anchura y la altura del primer bloque de croma respectivamente, o T y L representan la anchura y la altura del primer bloque de croma, respectivamente,

60 en donde N o T es un número entero par 2^n mayor o igual a un umbral (tal como 4, 8, 16, 32...) y n es un número entero positivo.

65 En una posible forma de implementación del método según cualquier implementación anterior del segundo aspecto o del segundo aspecto como tal,

– si el borde del bloque de croma es un borde de bloque de croma horizontal, la dirección a lo largo de la altura N del primer bloque de croma es perpendicular al borde del bloque de croma, y la dirección a lo largo de la altura T del segundo bloque de croma es perpendicular al borde del bloque de croma, en donde una o ambas alturas del primer y segundo bloque de croma son un número entero par 2ⁿ mayor que 4 (o mayor o igual a 8) o

– si el borde del bloque de croma es un borde de bloque de croma vertical, la dirección a lo largo de la anchura N del primer bloque de croma es perpendicular al borde del bloque, y la dirección a lo largo de la anchura T del segundo bloque de croma es perpendicular al borde del bloque, en donde una o ambas anchuras del primer y segundo bloque de croma son un número entero par 2ⁿ mayor que 4 (o mayor o igual a 8).

En una posible forma de implementación del método según cualquier implementación anterior del segundo aspecto o del segundo aspecto como tal, en donde el segundo bloque de imagen es un bloque actual y el primer bloque de imagen es un bloque vecino del bloque actual.

Solución 2:

Basado en el indicador JCCR, el QP de luma se correlaciona con el QP de croma para cada uno de los bloques P y Q por separado utilizando las tablas de mapeo de QP de croma respectivas y, a continuación, el promedio de los valores de QP de croma se establece como el valor de QP final que se usa para el método desbloqueo.

Si el bloque Q usa la herramienta Cb-Cr conjunta (JCCR), entonces $Q_{pcq} = \text{ChromaQpTable}[2][[Q_{pQ} + cQpPicOffset]$, de lo contrario $Q_{pcQ} = \text{ChromaQpTable}[cldx - 1][[Q_{pQ} + cQpPicOffset]$

Si el bloque P usa JCCR, entonces $Q_{pP} = \text{ChromaQpTable}[2][[Q_{pP} + cQpPicOffset]$; de lo contrario, $Q_{pP} = \text{ChromaQpTable}[cldx - 1][[Q_{pP} + cQpPicOffset]$

$$Q_{pC} = \text{Clip3}(0, 63, ((Q_{pCQ} + Q_{pCP} + 1) \gg 1)) \quad (8-1132')$$

Se puede entender que ChromaQpTable [2] representa la tabla de mapeo de QP de croma para un bloque codificado de Cb-Cr conjuntos (JCCR).

La etapa de determinar si el bloque P o el bloque Q o ambos bloques P y Q utilizan la herramienta JCCR puede incluir:

si la muestra $p_{0,0}$ está en una unidad de transformada con `tu_joint_cbcr_residual_flag` igual a 1, entonces se determina que el bloque P usa la herramienta JCCR; y/o

si la muestra $q_{0,0}$ está en una unidad de transformada con `tu_joint_cbcr_residual_flag` igual a 1, entonces se determina que el bloque Q usa la herramienta JCCR.

Los cambios en el texto de la especificación para la solución 2 se proporcionan con respecto a la sección 8.8.3.6.3 del documento JVET-O2001 (versión - vE).

8.8.3.6.3 Proceso de decisión para bordes en bloque de croma

Las variables Q_{pQ} y Q_{pP} se establecen iguales a los valores Q_{pY} de las unidades de codificación que incluyen los bloques de codificación que contienen la muestra $q_{0,0}$ y $p_{0,0}$, respectivamente.

La variable Q_{pc} se deriva del siguiente modo:

– Si la muestra $p_{0,0}$ o $q_{0,0}$ está en una unidad de transformada con `tu_joint_cbcr_residual_flag` igual a 1, entonces

o si la muestra $p_{0,0}$ está en una unidad de transformada con `tu_joint_cbcr_residual_flag` igual a 1, entonces

- $Q_{pP} = \text{ChromaQpTable}[2][[Q_{pP} + cQpPicOffset]$

o De lo contrario

- $Q_{pP} = \text{ChromaQpTable}[cldx - 1][[Q_{pP} + cQpPicOffset]$

o si la muestra $q_{0,0}$ está en una unidad de transformada con `tu_joint_cbcr_residual_flag` igual a 1, entonces

- $Q_{pQ} = \text{ChromaQpTable}[2][[Q_{pQ} + cQpPicOffset]$

o De lo contrario

- $Q_{pQ} = \text{ChromaQpTable}[cldx - 1][[Q_{pQ} + cQpPicOffset]$

$$Q_{pC} = \text{Clip3}(0, 63, ((Q_{pCp} + Q_{pCq} + 1) \gg 1)) \quad (8-1132)$$

- De lo contrario

$$q_{Pi} = \text{Clip3}(0, 63, ((Q_{pQ} + Q_{pP} + 1) \gg 1) + c_{QpPicOffset}) \quad (8-1132)$$

$$Q_{pC} = \text{ChromaQpTable}[cldx - 1][q_{Pi}] \quad (8-1133)$$

10 Puede entenderse que la variable Q_{pQ} representa el valor Q_{pY} para un bloque de luma Q correspondiente, y la variable Q_{pP} representa el valor de Q_{pY} para un bloque de luma P correspondiente.

15 Los efectos beneficiosos y las ventajas de la solución 2 según una realización de la presente invención son que el QP final obtenido para los bloques de JCCR es más preciso en comparación con la solución 1 y, por lo tanto, puede dar como resultado mejores decisiones de desbloqueo al resultar en una mejor calidad subjetiva.

Tercera realización de la presente invención

20 Según un tercer aspecto no cubierto por la invención reivindicada, este ejemplo se refiere a un método para de desbloqueo, para desbloquear un borde de bloque de croma entre un primer bloque de croma de un primer bloque de imagen y un segundo bloque de croma de un segundo bloque de imagen, en una codificación de imagen y/o una decodificación de imágenes,

en donde el método de desbloqueo comprende:

- realizar un proceso de decisión para el borde de bloque de croma, en donde el proceso de decisión comprende:
 - determinar un parámetro de cuantificación de croma Q_{pCp} para el primer bloque de croma basado en un primera QP de luma (tal como Q_{pYP}) de un primer bloque (801, 802) de luma del primer bloque de imagen usando una tabla de mapeo de Qp de croma (tal como ChromaQPTable) o entradas de información de la tabla de mapeo de Qp de croma que se indica mediante un índice (tal como índice de componente, cldx o índice de fila), y el índice (cldx) que tiene un primer valor de índice (tal como 3) cuando el primer bloque de croma es un bloque codificado de Cb-Cr conjuntos (JCCR), o el índice (cldx) que tiene un segundo valor de índice (tal como 1 o 2) cuando el primer bloque de croma es el componente de croma del primer bloque de imagen;
 - determinar un parámetro de cuantificación de croma Q_{pCq} para el segundo bloque de croma basado en un segundo QP de luma (tal como Q_{pYQ}) de un segundo bloque (802, 801) de luma del segundo bloque de imagen usando una tabla de mapeo de Qp de croma (tal como ChromaQPTable) o entradas de información de la tabla de mapeo de Qp de croma que se indica mediante un índice (tal como el índice de componente, cldx), y el índice (cldx) que tiene un primer valor (tal como 3) cuando el segundo bloque de croma es el bloque codificado de Cb-Cr conjuntos (JCCR), o el índice (cldx) que tiene un segundo valor (tal como 1 o 2) cuando el segundo bloque de croma es el componente de croma del segundo bloque de imagen; y
 - determinar un parámetro de cuantificación de croma Q_{pC} promediado y redondeado basado en el parámetro de cuantificación de croma Q_{pCp} para el primer bloque de croma y el parámetro de cuantificación de croma Q_{pCq} para el segundo bloque de croma;
- realizar un proceso de filtrado para el borde de bloque de croma basado en un resultado de decisión del proceso de decisión.

Puede entenderse que el resultado de la decisión indica que el borde de bloque debe filtrarse o no, y/o debe realizarse un filtrado prolongado o no.

55 En una posible forma de implementación del método según el tercer aspecto como tal, en donde diferentes valores de índice corresponden a diferentes entradas de información de una tabla de mapeo de QP de croma, o diferentes valores de índice corresponden a diferentes tablas de mapeo de QP de croma. En una posible forma de implementación del método según cualquier implementación anterior del tercer aspecto o del tercer aspecto como tal, en donde cuando en el caso de que el primer bloque de croma se codifique usando el modo de JCCR y el modo de JCCR sea un primer modo de JCCR (tal como 1),

60 determinar un parámetro de cuantificación del primer componente de croma (tal como el componente Cb) Q_{pCb} para el primer componente de croma (tal como el componente Cb) del primer bloque de imagen basado en un primer QP de luma (tal como el Q_{pYP}) de un primer bloque de luma del primer bloque de imagen usando una tabla de mapeo de Qp de croma cuyo índice (tal como el índice de componente, cldx) es el primer valor de índice o

usando una tabla de mapeo de Qp de croma que comprende entradas de información cuyo índice (tal como el índice de componente, cldx) es el primer valor de índice; y

5 determinar un parámetro de cuantificación del segundo componente de croma (tal como el componente Cr) QpCrp para el segundo componente de croma (tal como el componente Cr) del primer bloque de imagen basado en el parámetro de cuantificación del primer componente de croma (tal como el componente Cb) QpCb_p.

10 En una posible forma de implementación del método según cualquier implementación anterior del tercer aspecto o del tercer aspecto como tal, en donde en el caso de que el segundo bloque de croma se codifique usando el modo de JCCR y el modo de JCCR sea un primer modo de JCCR (tal como 1),

15 determinar un parámetro de cuantificación del primer componente de croma (tal como el componente Cb) QpCbq para el primer componente de croma (tal como el componente Cb) del segundo bloque de imagen basado en un segundo QP de luma (tal como Qp_{YQ}) de un segundo bloque de luma del segundo bloque de imagen usando una tabla de mapeo de Qp de croma cuyo índice (tal como el índice de componente, cldx) es el primer valor de índice o usando una tabla de mapeo de Qp de croma que comprende entradas de información cuyo índice (tal como el índice de componente, cldx) es el primer valor de índice; y

20 determinar un parámetro de cuantificación del segundo componente de croma (tal como el componente Cr) QpCrq para el segundo componente de croma (tal como el componente Cr) del segundo bloque de imagen basado en el parámetro de cuantificación del primer componente de croma (tal como el componente Cb) QpCbq. En una posible forma de implementación del método según cualquier implementación anterior del tercer aspecto o del tercer aspecto como tal, en donde cuando en el caso de que el primer bloque de croma se codifique usando el modo de JCCR y el modo de JCCR sea un segundo modo de JCCR (tal como 2),

25 determinar un parámetro de cuantificación del primer componente de croma (tal como el componente Cb) QpCb_p para el primer componente de croma (tal como el componente Cb) del primer bloque de imagen basado en un primer QP de luma (tal como el Qp_{YP}) de un primer bloque de luma del primer bloque de imagen usando una tabla de mapeo de Qp de croma cuyo índice (tal como el índice de componente, cldx) es el primer valor de índice o usando una tabla de mapeo de Qp de croma que comprende entradas de información cuyo índice (tal como el índice de componente, cldx) es el primer valor de índice; y

30 establecer un parámetro de cuantificación del segundo componente de croma (tal como el componente Cr) QpCrp para el segundo componente de croma (tal como el componente Cr) del primer bloque de imagen como el parámetro de cuantificación del primer componente de croma (tal como el componente Cb) QpCb_p.

35 En una posible forma de implementación del método según cualquier implementación anterior del tercer aspecto o del tercer aspecto como tal, en donde cuando en el caso de que el segundo bloque de croma se codifique usando el modo de JCCR y el modo de JCCR sea un segundo modo de JCCR (tal como 2),

40 determinar un parámetro de cuantificación del primer componente de croma (tal como el componente Cb) QpCbq para el primer componente de croma (tal como el componente Cb) del segundo bloque de imagen basado en un segundo QP de luma (tal como Qp_{YQ}) de un segundo bloque de luma de la segunda imagen usando una tabla de mapeo de Qp de croma cuyo índice (tal como el índice de componente, cldx) es el primer valor de índice o usando una tabla de mapeo de Qp de croma que comprende entradas de información cuyo índice (tal como el índice de componente, cldx) es el primer valor de índice; y

45 establecer un parámetro de cuantificación del segundo componente de croma (tal como el componente Cr) QpCrq para el segundo componente de croma (tal como el componente Cr) del segundo bloque de imagen como el parámetro de cuantificación del primer componente de croma (tal como el componente Cb) QpCbq.

50 En una posible forma de implementación del método según cualquier implementación anterior del tercer aspecto o del tercer aspecto como tal, en donde cuando en el caso de que el primer bloque de croma se codifique usando el modo de JCCR y el modo de JCCR sea un tercer modo de JCCR (tal como 3),

55 determinar un parámetro de cuantificación del segundo componente de croma (tal como el componente Cr) QpCrp para el segundo componente de croma (tal como el componente Cr) del primer bloque de imagen basado en un primer QP de luma (tal como Qp_{YP}) de un primer bloque de luma del primer bloque de imagen usando una tabla de mapeo de Qp de croma cuyo índice (tal como el índice de componente, cldx) es el primer valor de índice o usando una tabla de mapeo de Qp de croma que comprende entradas de información cuyo índice (tal como el índice de componente, cidx) es el primer valor de índice; y

60 determinar un parámetro de cuantificación del primer componente de croma (tal como el componente Cb) QpCb_p para el primer componente de croma (tal como el componente Cb) del primer bloque de imagen basado en el parámetro de cuantificación del segundo componente de croma (tal como el componente Cr) QpCrp.

65

En una posible forma de implementación del método según cualquier implementación anterior del tercer aspecto o del tercer aspecto como tal, en donde cuando en el caso de que el segundo bloque de croma se codifique usando el modo de JCCR y el modo de JCCR sea un tercer modo de JCCR (tal como 3),

5 determinar un parámetro de cuantificación del segundo componente de croma (tal como el componente Cr) $QpCrq$ para el segundo componente de croma (tal como el componente Cr) del segundo bloque de imagen basado en un segundo QP de luma (tal como Qp_{YQ}) de un segundo bloque de luma del segundo bloque de imagen usando una tabla de mapeo de Qp de croma cuyo índice (tal como el índice de componente, $cidx$) es el primer valor de índice o usando una tabla de mapeo de Qp de croma que comprende entradas de información cuyo índice (tal como el índice de componente, $cidx$) es el primer valor de índice; y

determinar un parámetro de cuantificación del primer componente de croma (tal como el componente Cb) $QpCbq$ para el primer componente de croma (tal como el componente Cb) del segundo bloque de imagen basado en el parámetro de cuantificación del segundo componente de croma (tal como el componente Cr) $QpCrq$. En una posible forma de implementación del método según cualquier implementación anterior del tercer aspecto o del tercer aspecto como tal, en donde la etapa de determinación de un parámetro de cuantificación de croma Qpc promediado y redondeado basado en el parámetro de cuantificación de croma Qp_{Cp} para el primer bloque de croma y el parámetro de cuantificación de croma Qp_{Cq} para el segundo bloque de croma, comprende:

20 determinar un parámetro de cuantificación del primer componente de croma (tal como el componente Cb) $QpCb$ promediado basado en

– el parámetro de cuantificación del primer componente de croma (tal como el componente Cb) $QpCb_p$ para el primer componente de croma (tal como el componente Cb) del primer bloque de imagen y

– el parámetro de cuantificación del primer componente de croma (tal como el componente Cb) $QpCb_q$ para el primer componente de croma (tal como el componente Cb) del segundo bloque de imagen.

En una posible forma de implementación del método según cualquier implementación anterior del tercer aspecto o del tercer aspecto como tal, en donde la etapa de determinación de un parámetro de cuantificación de croma Qpc promediado y redondeado basado en el parámetro de cuantificación de croma Qp_{Cp} para el primer bloque de croma y el parámetro de cuantificación de croma Qp_{Cq} para el segundo bloque de croma, comprende:

35 determinar un parámetro de cuantificación del segundo componente de croma (tal como el componente Cr) $QpCr$ promediado basado en

– el parámetro de cuantificación del segundo componente de croma (tal como el componente de Cr) $QpCr_p$ para el segundo componente de croma (tal como el componente de Cr) del primer bloque de imagen y

– el parámetro de cuantificación del segundo componente de croma (tal como el componente de Cr) $QpCr_q$ para el segundo componente de croma (tal como el componente de Cr) del segundo bloque de imagen.

En una posible forma de implementación del método según cualquier implementación anterior del tercer aspecto o del tercer aspecto como tal, en donde el parámetro de cuantificación de croma Qpc promediado y redondeado se usa para el proceso de decisión para el borde del bloque de croma, o el parámetro de cuantificación de croma Qpc promediado y redondeado se usa directa o indirectamente para determinar si el borde del bloque de croma debe filtrarse y/o si debe realizar un filtrado prolongado.

50 En una posible forma de implementación del método según cualquier implementación anterior del tercer aspecto o del tercer aspecto como tal, en donde

el valor de una primera variable de umbral (tal como β') se determina basada en el parámetro de cuantificación de croma Qpc promediado y redondeado o un primer Qpc recortado utilizando una tabla de mapeo; y el valor de una primera variable (tal como β) se deriva basada en el valor de la primera variable de umbral (tal como β');

en donde el valor de la primera variable (tal como β) se usa para el proceso de decisión para el borde del bloque de croma o se usa para determinar si el borde del bloque se debe filtrar y/o si se debe realizar un filtrado prolongado.

60 En una posible forma de implementación del método según cualquier implementación anterior del tercer aspecto o del tercer aspecto como tal, en donde

el valor de una segunda variable de umbral (tal como tc') se determina basado en el parámetro de cuantificación de croma Qpc promediado y redondeado o en un segundo Qpc recortado usando una tabla de mapeo; el valor de una segunda variable (tal como tc) se deriva basado en el valor de la segunda variable de umbral (tal como tc');

65

en donde el valor de la segunda variable (tal como t_c) se usa para el proceso de decisión para el borde del bloque de croma (o para determinar si el borde del bloque se debe filtrar y/o si se debe realizar un filtrado prolongado) y el proceso de filtrado para el borde del bloque de croma.

5 En una posible forma de implementación del método según cualquier implementación anterior del tercer aspecto o del tercer aspecto como tal, en donde la tabla de mapeo comprende el mapeo entre una pluralidad de las primeras variables de umbral, una pluralidad de las segundas variables de umbral y una pluralidad de parámetros de cuantificación.

10 En una posible forma de implementación del método según cualquier implementación anterior del tercer aspecto o del tercer aspecto como tal, en donde el primer bloque de croma es un primer componente de croma (tal como el componente Cb) del primer bloque de imagen y el segundo bloque de croma es un primer componente de croma (tal como el componente Cb) del segundo bloque de imagen, o

15 en donde el primer bloque de croma es un segundo componente de croma (tal como el componente Cr) del primer bloque de imagen y el segundo bloque de croma es un segundo componente de croma (tal como el componente Cr) del segundo bloque de imagen.

20 En una posible forma de implementación del método según cualquier implementación anterior del tercer aspecto o del tercer aspecto como tal, en donde el primer bloque de imagen y el segundo bloque de imagen son unidades de transformada o bloques de transformada.

25 En una posible forma de implementación del método según cualquier implementación anterior del tercer aspecto o del tercer aspecto como tal, en donde se determina si el al menos uno del primer bloque de croma y el segundo bloque (402, 502) de croma se codifica usando el modo de Cb-Cr conjuntos (JCCR) o si al menos uno del primer bloque de croma y el segundo bloque (402, 502) de croma es un bloque codificado de Cb-Cr conjuntos (JCCR) basado en un indicador de nivel de TU de unidad de transformada (tal como $tu_joint_cbr_residual_flag$).

30 En una posible forma de implementación del método según cualquier implementación anterior del tercer aspecto o del tercer aspecto como tal, en donde cuando el indicador de nivel de TU (tal como $tu_joint_cbr_residual_flag$) para el primer bloque de imagen es verdadero, el primer bloque de croma usa herramientas de codificación conjunta de residuos de crominancia (JCCR) o el primer bloque de croma se codifica usando el modo de Cb-Cr conjuntos (JCCR); o

35 cuando el indicador de nivel de TU (tal como $tu_joint_cbr_residual_flag$) para el segundo bloque de imagen es verdadero, el segundo bloque de croma usa herramientas de codificación conjunta de residuos de crominancia (JCCR) o el segundo bloque de croma se codifica usando el modo de Cb-Cr conjuntos (JCCR).

40 En una posible forma de implementación del método según cualquier implementación anterior del tercer aspecto o del tercer aspecto como tal, en donde la tabla de mapeo de Q_p de croma (tal como ChromaQPTable) comprende lo correspondiente entre una pluralidad de parámetros de cuantificación de croma Q_{p_c} y una pluralidad de parámetros de cuantificación de luma $Q_{p_{luma}}$, y la pluralidad de parámetros de cuantificación de croma Q_{p_c} se asocian con el primer valor de índice o el segundo valor de índice.

45 En una posible forma de implementación del método según cualquier implementación anterior del tercer aspecto o del tercer aspecto como tal, en donde el primer bloque de croma tiene un tamaño de bloque que es $M \times N$ o $N \times M$, y M y N representan la anchura y la altura del primer bloque de croma respectivamente, o N y M representan la anchura y la altura del primer bloque de croma respectivamente; en donde el segundo bloque de croma tiene un tamaño de bloque que es $L \times T$ o $T \times L$, y L y T representan la anchura y la altura del primer bloque de croma respectivamente, o T y L representan la anchura y la altura del primer bloque de croma, respectivamente,

50 en donde N o T es un número entero par 2^n mayor o igual a un umbral (tal como 4, 8, 16, 32...) y n es un número entero positivo.

55 En una posible forma de implementación del método según cualquier implementación anterior del tercer aspecto o del tercer aspecto como tal,

– si el borde del bloque de croma es un borde de bloque de croma horizontal, la dirección a lo largo de la altura N del primer bloque de croma es perpendicular al borde del bloque de croma, y la dirección a lo largo de la altura T del segundo bloque de croma es perpendicular al borde del bloque de croma, en donde una o ambas alturas del primer y segundo bloque de croma son un número entero par 2^n mayor que 4 (o mayor o igual a 8) o

60 – si el borde del bloque de croma es un borde de bloque de croma vertical, la dirección a lo largo de la anchura N del primer bloque de croma es perpendicular al borde del bloque, y la dirección a lo largo de la anchura T del segundo bloque de croma es perpendicular al borde del bloque, en donde una o ambas anchuras del primer y segundo bloque de croma son un número entero par 2^n mayor que 4 (o mayor o igual a 8).

65

En una posible forma de implementación del método según cualquier implementación anterior del tercer aspecto o del tercer aspecto como tal, en donde el segundo bloque de imagen es un bloque actual y el primer bloque de imagen es un bloque vecino del bloque actual.

5 Solución 3:

El valor de entrada de QP de croma depende del modo de JCCR.

Según el modo de JCCR, se determinan los valores de entrada de QP de croma.

10

⇒ Si el modo de JCCR == 1, entonces $QpCb = \text{ChromaQpTable}[2][Qpx]$ Y $QpCr = (QpCb + 1) \gg 1$

⇒ Si el modo de JCCR == 2, entonces $QpCb = \text{ChromaQpTable}[2][Qpx]$ Y $QpCr = QpCb$

15

⇒ Si el modo de JCCR == 3, entonces $QpCr = \text{ChromaQpTable}[2][Qpx]$ y $QpCb = (QpCr + 1) \gg 1$

* x en Qpx puede ser el valor de QP de un bloque P o Q.

20

$$QpCr = \text{Clip3}(0, 63, ((QpCr_Q + QpCr_P + 1) \gg 1)) \quad (8-1132a)$$

$$QpCb = \text{Clip3}(0, 63, ((QpCb_Q + QpCb_P + 1) \gg 1)) \quad (8-1132b)$$

25

Basada en TuCResMode, se determina la derivación de los valores de entrada de QP de croma para cada uno de los bloques P y Q.

Si TuCResMode == 1, la variable QPcb se deriva como $QPcb = \text{ChromaQpTable}[2][QP_x]$ y el valor QPCr se deriva como $QPCr = (QpCb + 1) \gg 1$

30

si no, si TuCResMode == 2, la variable QPcb se deriva como $QPcb = \text{ChromaQpTable}[2][QP_x]$ y el valor QPCr se deriva como $QPCr = QpCb$

si no, si TuCResMode == 3, la variable QPcr se deriva como $QPcr = \text{ChromaQpTable}[2][QP_x]$ y el valor QPCr se deriva como $QPCb = (QpCr + 1) \gg 1$

35

téngase en cuenta que x en QPx se puede reemplazar por P o Q para los bloques respectivos.

Los valores QPcr y QPcb se pueden derivar además del siguiente modo:

40

$$QpCr = \text{Clip3}(0, 63, ((QpCr_Q + QpCr_P + 1) \gg 1)) \quad (8-1132a)$$

$$QpCb = \text{Clip3}(0, 63, ((QpCb_Q + QpCb_P + 1) \gg 1)) \quad (8-1132b)$$

45

Los efectos beneficiosos y las ventajas de la solución 3, según una realización de la presente invención, son que el valor de QP usado por un bloque de croma (Cb o Cr) también se ajusta basado en el modo de JCCR (TuCResMode) y, por lo tanto, se puede obtener un QP más preciso para cada uno de los componentes Cb y Cr cuando se usa JCCR para señalar conjuntamente los componentes Cb-Cr.

Cuarta realización de la presente invención

50

Según un cuarto aspecto no cubierto por la invención reivindicada, este ejemplo se refiere a un método de desbloqueo para desbloquear un borde de bloque (903, 913, 923) de croma entre un primer bloque (901, 911, 921, 931, 941, 951, 961, 971) de croma de un primer bloque (601, 601') de imagen y un segundo bloque (902, 912, 922, 932, 942, 952, 962, 972) de croma de un segundo bloque (602, 602') de imagen, en una codificación de imagen y/o una decodificación de imagen,

55

en donde el método de desbloqueo comprende:

– realizar un proceso de decisión para el borde de bloque de croma, en donde el proceso de decisión comprende:

60

- determinar un parámetro de cuantificación de croma $QpCb_p$ para el primer bloque (901, 911, 921, 931, 941, 951, 961, 971) de croma basado en un primer QP de luma (tal como Qp_{VF}) para un primer bloque (801) de luma del primer bloque (601, 601') de imagen y una regla de conversión, en donde la regla de conversión es una primera regla de conversión cuando el primer bloque (901, 911, 921, 931, 941, 951, 961, 971) de croma es un bloque codificado Joint Cb-Cr (JCCR), o la regla de conversión es una segunda regla de conversión cuando el primer bloque (901, 911, 921, 931, 941, 951, 961, 971) de croma es el componente de croma del primer bloque (601, 601') de imagen;

65

- 5 - determinar un parámetro de cuantificación de croma Q_{pcq} para el segundo bloque (902, 912, 922, 932, 942, 952, 962, 972) de croma basado en un segundo QP de luma (tal como $Q_{p_{\gamma Q}}$) de un segundo bloque (802) de luma del segundo bloque (602, 602') de imagen y una regla de conversión, en donde la regla de conversión es una primera regla de conversión cuando el segundo bloque de croma es un bloque codificado Joint Cb-Cr (JCCR), o la regla de conversión es una segunda regla de conversión cuando el segundo bloque de croma es el componente de croma del segundo bloque (602, 602') de imagen; y
- 10 - determinar un parámetro de cuantificación de croma Q_{pc} promediado y redondeado basado en el parámetro de cuantificación de croma Q_{pcp} para el primer bloque (901, 911, 921, 931, 941, 951, 961, 971) de croma y el parámetro de cuantificación de croma Q_{pcq} para el segundo bloque (902, 912, 922, 932, 942, 952, 962, 972) de croma;
- realizar un proceso de filtrado para el borde (903, 913, 923) del bloque de croma basado en un resultado de decisión del proceso de decisión.
- 15 En una posible forma de implementación del método según el cuarto aspecto como tal, la primera regla de conversión se diseña para un bloque codificado Joint Cb-Cr (JCCR).
- 20 En una posible forma de implementación del procedimiento según cualquier implementación anterior del cuarto aspecto o del cuarto aspecto como tal, la segunda regla de conversión comprende una segunda regla R21 de conversión y/o una segunda regla R22 de conversión,
- 25 en donde la segunda regla R21 de conversión y la segunda regla R22 de conversión se diseñan para un primer componente de croma (tal como un componente Cb) y un segundo componente de croma (tal como un componente de Cr) respectivamente.
- 30 En una posible forma de implementación, la primera regla de conversión es diferente de una segunda regla R21 de conversión y una segunda regla R22 de conversión.
- 35 En una posible forma de implementación del método según cualquier implementación anterior del cuarto aspecto o del cuarto aspecto como tal, la regla de conversión es una segunda regla R21 de conversión cuando el primer bloque de croma es un primer componente (911) de croma del primer bloque (601, 601') de imagen, y/o la regla de conversión es una segunda regla R22 de conversión cuando el primer bloque de croma es un segundo componente (921) de croma del primer bloque (601, 601') de imagen;
- 40 o la regla de conversión es una segunda regla R21 de conversión cuando el segundo bloque de croma es un primer componente (912) de croma del segundo bloque (602, 602') de imagen, y/o la regla de conversión es una segunda regla R22 de conversión cuando el segundo bloque de croma es un segundo componente (922) de croma del segundo bloque (602, 602') de imagen.
- 45 En una posible forma de implementación del procedimiento según cualquier implementación anterior del cuarto aspecto o del cuarto aspecto como tal, la primera regla de conversión, la segunda regla R21 de conversión y la segunda regla R22 de conversión corresponden a (o están representadas por) las entradas de información respectivas de una tabla de mapeo de Q_p de croma (tal como ChromaQPTable), o la primera regla de conversión, la segunda regla R21 de conversión
- 50 y la segunda regla R22 de conversión corresponden a (o están representadas por) una primera tabla de mapeo de Q_p de croma (tal como ChromaQPTable), una segunda tabla de mapeo de Q_p de croma y una tercera tabla de mapeo de Q_p de croma, respectivamente.
- 55 En una posible forma de implementación del método según cualquier implementación anterior del cuarto aspecto o del cuarto aspecto como tal, la primera regla de conversión, la segunda regla R21 de conversión y la segunda regla R22 de conversión corresponden a una primera fórmula, una segunda fórmula y una tercera fórmula, respectivamente;
- 60 o la primera regla de conversión, la segunda regla R21 de conversión y la segunda regla R22 de conversión son una primera fórmula, una segunda fórmula y una tercera fórmula, respectivamente.
- 65 En una posible forma de implementación del método según cualquier implementación anterior del cuarto aspecto o del cuarto aspecto como tal, la primera regla de conversión se representa en términos de una tabla de mapeo de Q_p de croma (tal como ChromaQPTable) y una primera fórmula;
- la segunda regla R21 de conversión se representa en términos de una tabla de mapeo de Q_p de croma (tal como ChromaQPTable) y una segunda fórmula;

la segunda regla R22 de conversión se representa en términos de una tabla de mapeo de Qp de croma (tal como ChromaQPTable) y una tercera fórmula.

5 En una posible forma de implementación del método según cualquier implementación anterior del cuarto aspecto o del cuarto aspecto como tal, la primera regla de conversión se representa en términos de una tabla de mapeo de Qp de croma (tal como ChromaQPTable) y un primer conjunto de fórmulas;

10 la segunda regla R21 de conversión se representa en términos de una tabla de mapeo de Qp de croma (tal como ChromaQPTable) y un segundo conjunto de fórmulas;

10 la segunda regla R22 de conversión se representa en términos de una tabla de mapeo de Qp de croma (tal como ChromaQPTable) y un tercer conjunto de fórmulas.

15 En una posible forma de implementación del procedimiento según cualquier implementación anterior del cuarto aspecto o del cuarto aspecto como tal, la primera regla de conversión, la segunda regla R21 de conversión y la segunda regla R22 de conversión se indican mediante un primer valor de índice, un segundo valor de índice y un tercer valor de índice, respectivamente, o

20 la primera regla de conversión, la segunda regla R21 de conversión y la segunda regla R22 de conversión corresponden a un primer valor de índice, un segundo valor de índice y un tercer valor de índice, respectivamente.

En una posible forma de implementación, el primer valor de índice, el segundo valor de índice y el tercer valor de índice son diferentes entre sí.

25 En una posible forma de implementación del procedimiento según cualquier implementación anterior del cuarto aspecto o del cuarto aspecto como tal, el primer valor de índice, el segundo valor de índice y el tercer valor de índice son valores enteros diferentes.

30 En una posible forma de implementación del método según cualquier implementación anterior del cuarto aspecto o del cuarto aspecto como tal, el primer valor de índice, el segundo valor de índice y el tercer valor de índice son índices de componente (tales como cldx) que tienen un valor diferente que especifica el bloque codificado Cb-Cr conjunto, el componente Cb y el componente Cr, respectivamente.

35 En una posible forma de implementación del procedimiento según cualquier implementación anterior del cuarto aspecto o del cuarto aspecto como tal, el primer valor de índice es 3 o 4 o 5, el segundo valor de índice es 1 y el tercer índice es 2.

40 En una posible forma de implementación del método según cualquier implementación anterior del cuarto aspecto o del cuarto aspecto como tal, el parámetro de cuantificación de croma Q_{p_c} promediado y redondeado se usa para el proceso de decisión para el borde del bloque de croma; o

el parámetro de cuantificación de croma Q_{p_c} promediado y redondeado se usa directa o indirectamente para determinar si el borde (903, 913, 923) del bloque de croma debe filtrarse y/o si debe realizarse un filtrado prolongado.

45 En una posible forma de implementación del método según cualquier implementación anterior del cuarto aspecto o del cuarto aspecto como tal,

el valor de una primera variable de umbral (tal como (3')) se determina basado en el parámetro de cuantificación de croma Q_{p_c} promediado y redondeado o un primer Q_{p_c} recortado usando una tabla de consulta; y

50 el valor de una primera variable (tal como β) se obtiene basado en el valor de la primera variable de umbral (tal como (3'));

en donde el valor de la primera variable (tal como β) se usa para el proceso de decisión para el borde del bloque de croma o el valor de la primera variable (tal como β) se usa para determinar si el borde (903, 913, 923) del bloque de croma se debe filtrar y/o si se debe realizar un filtrado prolongado.

55 En una posible forma de implementación del método según cualquier implementación anterior del cuarto aspecto o del cuarto aspecto como tal,

60 el valor de una segunda variable de umbral (tal como tc') se determina basado en el parámetro de cuantificación de croma Q_{p_c} promediado y redondeado o un segundo Q_{p_c} recortado usando una tabla de consulta;

el valor de una segunda variable (tal como tc) se deriva basado en el valor de la segunda variable de umbral (tal como tc');

65 en donde el valor de la segunda variable (tal como tc) se usa para el proceso de decisión para el borde del bloque de croma (o el valor de la segunda variable (tal como tc) se usa para determinar si el borde (903, 913, 923) del bloque de croma se debe filtrar y/o si se debe realizar un filtrado prolongado) y el proceso de filtrado para el borde del bloque de croma.

En una posible forma de implementación del método según cualquier implementación anterior del cuarto aspecto o del cuarto aspecto como tal, la tabla de consulta comprende el mapeo entre una pluralidad de las primeras variables de umbral, una pluralidad de las segundas variables de umbral y una pluralidad de parámetros de cuantificación.

5 En una posible forma de implementación del método según cualquier implementación anterior del cuarto aspecto o del cuarto aspecto como tal, el primer bloque de croma es un primer componente de croma (tal como el componente Cb) del primer bloque de imagen y el segundo bloque de croma es un primer componente de croma (tal como el componente Cb) del segundo bloque de imagen, y/o

10 en donde el primer bloque de croma es un segundo componente de croma (tal como el componente Cr) del primer bloque de imagen y el segundo bloque de croma es un segundo componente de croma (tal como el componente Cr) del segundo bloque de imagen.

15 En una posible forma de implementación del método según cualquier implementación anterior del cuarto aspecto o del cuarto aspecto como tal, al menos uno del primer componente de croma (tal como el componente Cb) del primer bloque de imagen y el primer componente de croma (tal como el componente Cb) del segundo bloque de imagen se codifican usando el modo de Cb-Cr conjuntos (JCCR), o en donde al menos uno del segundo componente de croma (tal como el componente Cr) del primer bloque de imagen y el segundo componente de croma (tal como el componente Cr) del segundo bloque de imagen se codifican usando el modo de Cb-Cr conjuntos (JCCR).

20 En una posible forma de implementación del método según cualquier implementación anterior del cuarto aspecto o del cuarto aspecto como tal, el primer bloque de imagen y el segundo bloque de imagen son bloques de transformada; o el primer bloque de imagen y el segundo bloque de imagen son bloques de codificación.

25 En una posible forma de implementación del método según cualquier implementación anterior del cuarto aspecto o del cuarto aspecto como tal, se determina si el al menos uno del primer bloque (901, 911, 921, 931, 941, 951, 961, 971) de croma y el segundo bloque (902, 912, 922, 932, 942, 952, 962, 972) de croma se codifica usando el modo de Cb-Cr conjuntos (JCCR) o el al menos uno del primer bloque (901, 911, 921, 931, 941, 951, 961, 971) de croma y el segundo bloque (902, 912, 922, 932, 942, 952, 962, 972) de croma es un bloque codificado de Cb-Cr conjuntos (JCCR) basado en un indicador de nivel de TU (tal como `tu_joint_cbcr_residual_flag`) de la unidad de transformada.

30 En una posible forma de implementación del método según cualquier implementación anterior del cuarto aspecto o del cuarto aspecto como tal, cuando el indicador de nivel de TU (tal como `tu_joint_cbcr_residual_flag`) para el primer bloque de imagen es verdadero, el primer bloque de croma es un bloque codificado de Cb-Cr conjuntos (JCCR) o el primer bloque de croma se codifica usando el modo de Cb-Cr conjuntos (JCCR); o cuando el indicador de nivel de TU (tal como `tu_joint_cbcr_residual_flag`) para el segundo bloque de imagen es verdadero, el segundo bloque de croma es un bloque codificado de Cb-Cr conjuntos (JCCR) o el segundo bloque de croma se codifica usando el modo de Cb-Cr conjuntos (JCCR).

35 En una posible forma de implementación del método según cualquier implementación anterior del cuarto aspecto o del cuarto aspecto como tal,

40 en donde el primer bloque (901, 911, 921, 931, 941, 951, 961, 971) de croma tiene un tamaño de bloque que es $M \times N$ o $N \times M$, y M y N representan la anchura y la altura del primer bloque de croma respectivamente, o N y M representan la anchura y la altura del primer bloque de croma respectivamente; en donde el segundo bloque (902, 912, 922, 932, 942, 952, 962, 972) de croma tiene un tamaño de bloque que es $L \times T$ o $T \times L$, y L y T representan la anchura y la altura del primer bloque de croma, respectivamente, o T y L representan la anchura y la altura del primer bloque de croma, respectivamente, en donde N o T es un número entero par, 2^n (tal como 4, 8, 16 o 32) y n es un número entero positivo.

45 En una posible forma de implementación,

50 – si el borde (903, 913, 923) del bloque de croma es un borde de bloque de croma horizontal, la dirección a lo largo de la altura N del primer bloque de croma es perpendicular al borde del bloque de croma, y la dirección a lo largo de la altura T del segundo bloque de croma es perpendicular al borde del bloque de croma; o

55 – si el borde (903, 913, 923) del bloque de croma es un borde (903, 913, 923) de bloque de croma vertical, la dirección a lo largo de la anchura N del primer bloque de croma es perpendicular al borde (903, 913, 923) del bloque, y la dirección a lo largo de la anchura T del segundo bloque de croma es perpendicular al borde (903, 913, 923) del bloque.

60 En una posible forma de implementación del método según cualquier implementación anterior del cuarto aspecto o del cuarto aspecto como tal, el segundo bloque de imagen es un bloque actual y el primer bloque de imagen es un bloque vecino del bloque actual.

65

En una posible forma de implementación del método según cualquier implementación anterior del cuarto aspecto o del cuarto aspecto como tal, cuando en el caso de que el primer bloque de croma se codifique usando el modo de JCCR y el modo de JCCR sea un primer modo de JCCR (tal como 1),

5 determinar un parámetro de cuantificación del primer componente de croma (tal como el componente Cb) QpCbp para el primer componente de croma (tal como el componente Cb) del primer bloque de imagen basado en un primer QP de luma (tal como Qp_{YF}) de un primer bloque (801) de luma del primer bloque de imagen y la primera regla de conversión; y

10 determinar un parámetro de cuantificación del segundo componente de croma (tal como el componente Cr) QpCrp para el segundo componente de croma (tal como el componente Cr) del primer bloque de imagen basado en el parámetro de cuantificación del primer componente de croma (tal como el componente Cb) QpCbp.

15 En una posible forma de implementación del método según cualquier implementación anterior del cuarto aspecto o del cuarto aspecto como tal, cuando en el caso de que el segundo bloque de croma se codifique usando el modo de JCCR y el modo de JCCR sea un primer modo de JCCR (tal como 1),

20 determinar un parámetro de cuantificación del primer componente de croma (tal como el componente Cb) QpCbq para el primer componente de croma (tal como el componente Cb) del segundo bloque de imagen basado en un segundo QP de luma (tal como Qp_{YO}) de un segundo bloque (802) de luma del segundo bloque de imagen y la primera regla de conversión; y

25 determinar un parámetro de cuantificación del segundo componente de croma (tal como el componente Cr) QpCrq para el segundo componente de croma (tal como el componente Cr) del segundo bloque de imagen basado en el parámetro de cuantificación del primer componente de croma (tal como el componente Cb) QpCbq. En una posible forma de implementación del método según cualquier implementación anterior del cuarto aspecto o del cuarto aspecto como tal, si el modo de JCCR == 1, entonces QpCb = ChromaQpTable [2] [Qpx], o si el modo de JCCR == 1, entonces QpCb = ChromaQpTable [cldx - 1] [Qpx],, cldx=3;

30 y

$$QpCr = (QpCb + 1) \gg 1,$$

35 en donde Qpx representa un primer QP de luma (tal como Qp_{YF}) del primer bloque de luma del primer bloque de imagen, o un segundo QP de luma (tal como Qp_{YO}) del segundo bloque de luma del segundo bloque de imagen; en donde ChromaQpTable representa una tabla de mapeo de QP de croma cuyo índice es 2 o una tabla de mapeo de QP de croma que comprende entradas de información cuyo índice es 2; o en donde ChromaQpTable representa una tabla de mapeo de QP de croma cuyo índice es 3 o una tabla de mapeo de QP de croma que comprende entradas de información cuyo índice es 3.

40 En una posible forma de implementación del método según cualquier implementación anterior del cuarto aspecto o del cuarto aspecto como tal, cuando en el caso de que el primer bloque de croma se codifique usando el modo de JCCR y el modo de JCCR sea un segundo modo de JCCR (tal como 2),

45 determinar un parámetro de cuantificación del primer componente de croma (tal como el componente Cb) QpCbp para el primer componente de croma (tal como el componente Cb) del primer bloque de imagen basado en un primer QP de luma (tal como Qp_{YF}) de un primer bloque (801) de luma del primer bloque de imagen y la primera regla de conversión; y

50 establecer un parámetro de cuantificación del segundo componente de croma (tal como el componente Cr) QpCrp para el segundo componente de croma (tal como el componente Cr) del primer bloque de imagen como el parámetro de cuantificación del primer componente de croma (tal como el componente Cb) QpCbp.

55 En una posible forma de implementación del método según cualquier implementación anterior del cuarto aspecto o del cuarto aspecto como tal, cuando en el caso de que el segundo bloque de croma se codifique usando el modo de JCCR y el modo de JCCR sea un segundo modo de JCCR (tal como 2),

determinar un parámetro de cuantificación del primer componente de croma (tal como el componente Cb) QpCbq para el primer componente de croma (tal como el componente Cb) del segundo bloque de imagen basado en un segundo QP de luma de un segundo bloque (802) de luma del segundo bloque de imagen y la primera regla de conversión; y

60 establecer un parámetro de cuantificación del segundo componente de croma (tal como el componente Cr) QpCrq para el segundo componente de croma (tal como el componente Cr) del segundo bloque de imagen como el parámetro de cuantificación del primer componente de croma (tal como el componente Cb) QpCbq.

65 En una posible forma de implementación del método según cualquier implementación anterior del cuarto aspecto o del cuarto aspecto como tal, si el modo de JCCR == 2, entonces QpCb = ChromaQpTable [2] [Qpx],, o si el modo de JCCR == 2, entonces QpCb = ChromaQpTable [cldx - 1] [Qpx],, cldx=3;

y

$$QpCr = QpCb,$$

en donde Qpx representa un primer QP de luma del primer bloque de luma del primer bloque de imagen, o un segundo QP de luma del segundo bloque de luma del segundo bloque de imagen;

en donde ChromaQpTable representa una tabla de mapeo de QP de croma cuyo índice es 2 o una tabla de mapeo de QP de croma que comprende entradas de información cuyo índice es 2; o en donde ChromaQpTable representa una tabla de mapeo de QP de croma cuyo índice es 3 o una tabla de mapeo de QP de croma que comprende entradas de información cuyo índice es 3.

En una posible forma de implementación del método según cualquier implementación anterior del cuarto aspecto o del cuarto aspecto como tal, cuando en el caso de que el primer bloque de croma se codifique usando el modo de JCCR y el modo de JCCR sea un tercer modo de JCCR (tal como 3),

determinar un parámetro de cuantificación del segundo componente de croma (tal como el componente Cr) QpCrp para el segundo componente de croma (tal como el componente Cr) del primer bloque de imagen basado en un primer QP de luma de un primer bloque (801) de luma del primer bloque de imagen y la primera regla de conversión; y

determinar un parámetro de cuantificación del primer componente de croma (tal como el componente Cb) QpCbq para el primer componente de croma (tal como el componente Cb) del primer bloque de imagen basado en el parámetro de cuantificación del segundo componente de croma (tal como el componente Cr) QpCrp.

En una posible forma de implementación del método según cualquier implementación anterior del cuarto aspecto o del cuarto aspecto como tal, cuando en el caso de que el segundo bloque de croma se codifique usando el modo de JCCR y el modo de JCCR sea un tercer modo de JCCR (tal como 3),

determinar un parámetro de cuantificación del segundo componente de croma (tal como el componente Cr) QpCrq para el segundo componente de croma (tal como el componente Cr) del segundo bloque de imagen basado en un segundo QP de luma de un segundo bloque (802) de luma del segundo bloque de imagen y la primera regla de conversión; y

determinar un parámetro de cuantificación del primer componente de croma (tal como el componente Cb) QpCbq para el primer componente de croma (tal como el componente Cb) del segundo bloque de imagen basado en el parámetro de cuantificación del segundo componente de croma (tal como el componente Cr) QpCrq. En una posible forma de implementación del método según cualquier implementación anterior del cuarto aspecto o del cuarto aspecto como tal, si el modo de JCCR == 3, entonces $QpCr = ChromaQpTable [2] [Qpx],$ o si el modo de JCCR == 3, entonces $QpCr = ChromaQpTable [cidx - 1] [Qpx],$ cidx=3;

y

$$y QpCb = (QpCr + 1) \gg 1,$$

en donde Qpx representa un primer QP de luma del primer bloque de luma del primer bloque de imagen, o un segundo QP de luma del segundo bloque de luma del segundo bloque de imagen;

en donde ChromaQpTable representa una tabla de mapeo de QP de croma cuyo índice es 2 o una tabla de mapeo de QP de croma que comprende entradas de información cuyo índice es 2; o en donde ChromaQpTable representa una tabla de mapeo de QP de croma cuyo índice es 3 o una tabla de mapeo de QP de croma que comprende entradas de información cuyo índice es 3.

En una posible forma de implementación del método según cualquier implementación anterior del cuarto aspecto o del cuarto aspecto como tal, la etapa de determinación de un parámetro de cuantificación Qpc promediado y redondeado basado en el parámetro de cuantificación de croma Qp_{Cb} para el primer bloque de croma y el parámetro de cuantificación de croma Qp_{Cr} para el segundo bloque de croma, comprende:

determinar un parámetro de cuantificación del primer componente de croma (tal como el componente Cb) QpCb promediado y redondeado basado en

– el parámetro de cuantificación del primer componente de croma (tal como el componente Cb) QpCbq para el primer componente de croma (tal como el componente Cb) del primer bloque de imagen y

– el parámetro de cuantificación del primer componente de croma (tal como el componente Cb) QpCbq para el primer componente de croma (tal como el componente Cb) del segundo bloque de imagen.

En una posible forma de implementación del método según cualquier implementación anterior del cuarto aspecto o del cuarto aspecto como tal, la etapa de determinación de un parámetro de cuantificación Q_{pc} promediado y redondeado basado en el parámetro de cuantificación de croma $Q_{p_{Cb}}$ para el primer bloque de croma y el parámetro de cuantificación de croma $Q_{p_{Cr}}$ para el segundo bloque de croma, comprende:

5 determinar un parámetro de cuantificación del segundo componente de croma (tal como el componente Cr) Q_{pCr} promediado y redondeado basado en

– el parámetro de cuantificación del segundo componente de croma (tal como el componente de Cr) Q_{pCrp} para el segundo componente de croma (tal como el componente de Cr) del primer bloque de imagen y

– el parámetro de cuantificación del segundo componente de croma (tal como el componente de Cr) Q_{pCrq} para el segundo componente de croma (tal como el componente de Cr) del segundo bloque de imagen.

15 Solución 4:

Según el tipo de bloque de croma, es decir, Cb, Cr o Cb-Cr (componente Cb, componente de Cr o bloque codificado de Cb-Cr conjunto), y el modo de Cb-Cr conjuntos para el tipo de bloque Cb-Cr es igual a 2, es decir, si $TuCResMode[xCb][yCb]$ del bloque dado es igual a 2, como se muestra en la ecuación 8-952 a continuación, el QP de luma se asigna al QP de croma para cada uno de los bloques P y Q por separado utilizando una regla de conversión respectiva (p. ej., una combinación de una tabla de mapeo de QP de croma y fórmulas respectivas) y, a continuación, el promedio de los valores de QP de croma de los bloques P y Q se establece como el valor de QP final, que después se usa para otras operaciones desbloqueo.

25 Un ejemplo de las reglas de conversión es el siguiente: la regla de conversión para el componente Cb se muestra más adelante en la sección 8.7.1, utilizando la fórmula 8-935, 8-936 y 8-939; En otras palabras, las ecuaciones 8-935, 8-936 y 8-939 son un ejemplo de la segunda regla R21 de conversión que se diseña para un primer componente de croma (tal como un componente Cb). La ecuación 8-936 se representa en términos de una tabla de mapeo de QP de croma.

30 La regla de conversión para el componente Cr se indica más adelante en la sección 8.7.1, utilizando las fórmulas 8-935, 8-937 y 8-940; En otras palabras, las ecuaciones 8-935, 8-937 y 8-940 son un ejemplo de la segunda regla R22 de conversión que se diseña para un segundo componente de croma (tal como un componente de Cr). La ecuación 8-937 se representa en términos de una tabla de mapeo de QP de croma. La regla de conversión para el componente CbCr (es decir, el componente Cb-Cr conjunto) se indica más adelante en la sección 8.7.1, utilizando las fórmulas 8-935, 8-938 y 8-941; En otras palabras, las ecuaciones 8-935, 8-938 y 8-941 son un ejemplo de la primera regla de conversión que se diseña para una componente de Cb-Cr conjunto. La ecuación 8-938 se representa en términos de una tabla de mapeo de QP de croma. Téngase en cuenta que la presente descripción propone usar diferentes reglas de conversión para diferentes componentes de croma, pero no se limita a las reglas particulares o fórmulas particulares mencionadas en la sección 8.7.1.

40 Los cambios exactos en el texto de la especificación para la solución 4 son los que se indican a continuación.

8.8.3.6.3 Proceso de decisión para bordes de bloques de croma

Este proceso solo se invoca cuando $ChromaArrayType$ no es igual a 0.

45 Las entradas de este proceso son:

– una matriz de muestra de imagen de croma $recPicture$,

50 – una ubicación de croma (xCb , yCb) que especifica la muestra superior izquierda del bloque de codificación de croma actual en relación con la muestra de croma superior izquierda de la imagen actual,

– una ubicación de croma (xBI , yBI) que especifica la muestra superior izquierda del bloque de croma actual en relación con la muestra superior izquierda del bloque de codificación de croma actual,

55 – una variable $edgeType$ que especifica si se filtra un borde vertical (EDGE_VER) u horizontal (EDGE_HOR),

– una variable $cIdx$ que especifica el índice del componente de color,

60 – una variable $cQpPicOffset$ que especifica el desplazamiento del parámetro de cuantificación de croma a nivel de imagen,

– una variable bS que especifica la intensidad de filtrado de límites,

65 – una variable $maxFilterLengthCbCr$.

Las salidas de este proceso son

- 5 – la variable modificada $\text{maxFilterLengthCbCr}$,
- la variable t_c .

La variable maxK se deriva del siguiente modo:

- 10 – Si edgeType es igual a EDGE_VER , se aplica lo siguiente:

$$\text{maxK} = (\text{SubHeightC} = 1) ? 3 : 1 \quad (8-1124)$$

- 15 – De lo contrario (edgeType es igual a EDGE_HOR), se aplica lo siguiente:

$$\text{maxK} = (\text{SubWidthC} = 1) ? 3 : 1 \quad (8-1125)$$

Los valores p_i y q_i con $i = 0.. \text{maxFilterLengthCbCr}$ y $k = 0.. \text{maxK}$ se derivan del siguiente modo:

- 20 – Si edgeType es igual a EDGE_VER , se aplica lo siguiente:

$$q_{i,k} = \text{recPicture}[\text{xCb} + \text{xBI} + i] [\text{yCb} + \text{yBI} + k] \quad (8-1126)$$

$$P_{i,k} = \text{recPicture}[\text{xCb} + \text{xBI} - i - 1] [\text{yCb} + \text{yBI} + k] \quad (8-1127)$$

$$\text{subSampleC} = \text{SubHeightC} \quad (8-1128)$$

30 la variable Q_{pQ} se obtiene invocando el proceso de derivación de QP de croma tal como se especifica en la cláusula 8.8.3.6.10 con la ubicación de croma ($\text{xCb} + \text{xBI} + i$, $\text{yCb} + \text{yBI} + k$) y cldx como entradas// Aquí, Q_{pQ} es el valor de QP de croma, para distinguir el valor de QP de luma y el valor de QP de croma, se representará mediante Q_{pcQ} en la parte reivindicación//

35 la variable Q_{pP} se obtiene invocando el proceso de derivación de QP de croma tal como se especifica en la cláusula 8.8.3.6.10 con la ubicación de croma ($\text{xCb} + \text{xBI} - i - 1$, $\text{yCb} + \text{yBI} + k$) y cldx como entradas// Aquí, Q_{pP} es el valor de QP de croma, para distinguir el valor de QP de luma y el valor de QP de croma, se representará mediante Q_{pcP} en la parte reivindicación //

- 40 – De lo contrario (edgeType es igual a EDGE_HOR), se aplica lo siguiente:

$$q_{i,k} = \text{recPicture}[\text{xCb} + \text{xBI} + k] [\text{yCb} + \text{yBI} + i] \quad (8-1129)$$

$$P_{i,k} = \text{recPicture}[\text{xCb} + \text{xBI} + k] [\text{yCb} + \text{yBI} - i - 1] \quad (8-1130)$$

$$\text{subSampleC} = \text{SubWidthC} \quad (8-1131)$$

45 la variable Q_{pQ} se obtiene invocando el proceso de derivación de QP de croma tal como se especifica en la cláusula 8.8.3.6.10 con la ubicación de croma ($\text{xCb} + \text{xBI} + k$, $\text{yCb} + \text{yBI} + i$) y cldx como entradas// Q_{pQ} es el valor de QP de croma, que se representará mediante Q_{pcQ} en la parte reivindicaciones//

50 la variable Q_{pP} se obtiene invocando el proceso de derivación de QP de croma tal como se especifica en la cláusula 8.8.3.6.10 con la ubicación de croma ($\text{xCb} + \text{xBI} + k$, $\text{yCb} + \text{yBI} - i - 1$) y cldx como entradas// Q_{pP} es el valor de QP de croma, que se representará mediante Q_{pcP} en la parte reivindicaciones //

55 La variable Q_{pc} se deriva del siguiente modo:

$$q_{p_i} = \text{Clip3}(0, 63, ((Q_{pQ} + Q_{pP} + 1) >> 1)) \quad (8-1132)$$

60 El valor de la variable β^i se determina tal como se especifica en la Tabla 8-18 basado en el parámetro de cuantificación Q obtenido del siguiente modo:

$$Q = \text{Clip3}(0, 63, Q_{pc} + (\text{slice_beta_offset_div2} << 1)) \quad (8-1134)$$

65 donde $\text{slice_beta_offset_div2}$ es el valor del elemento de sintaxis $\text{slice_beta_offset_div2}$ para el segmento que contiene la muestra $q_{0,0}$.

La variable β se deriva del siguiente modo:

$$\beta = \beta' * (1 \ll (\text{BitDepth}_c - 8)) \quad (8-1135)$$

5 El valor de la variable t_c se determina tal como se especifica en la Tabla 8-18 basado en el parámetro de cuantificación Q de croma obtenido del siguiente modo:

$$Q = \text{Clip3}(0, 65, Q_p + 2 * (bS - 1) + (\text{slice_tc_offset_div2} \ll 1)) \quad (8-1136)$$

10 donde $\text{slice_tc_offset_div2}$ es el valor del elemento de sintaxis $\text{slice_tc_offset_div2}$ para el segmento que contiene la muestra $q_{0,0}$.

La variable t_c se deriva del siguiente modo:

$$15 \quad t_c = (\text{BitDepth}_c < 10) ? (t_c' + 2) \ll (10 - \text{BitDepth}_c) : t_c' * (1 \ll (\text{BitDepth}_c - 8)) \quad (8-1137)$$

8.8.3.6.10 Proceso de derivación de parámetros de cuantificación para bloques de codificación de croma

Este proceso solo se invoca cuando ChromaArrayType no es igual a 0.

20

Las entradas de este proceso son:

- Un bloque de codificación de croma que contiene una ubicación de muestra de croma dada (x_{Cb} , y_{Cb})
 - 25 – una variable $cldx$ que especifica el índice del componente de color del bloque de codificación de croma dado,
- la salida de este proceso es un parámetro de cuantificación q_P del bloque de codificación que contiene la muestra (x_{Cb} , y_{Cb}).

30 – Si $\text{TuCResMode}[x_{Cb}][y_{Cb}]$ es igual a 2, se aplica lo siguiente:

$$q_P = Q_{p'_{cbCr}} - Q_{pBdOffset}_c \quad (8-952)$$

35 – De lo contrario, si $cldx$ es igual a 1, se aplica lo siguiente:

$$q_P = Q_{p'_{cb}} - Q_{pBdOffset}_c \quad (8-953)$$

– De lo contrario ($cldx$ es igual a 2), se aplica lo siguiente:

$$40 \quad q_P = Q_{p'_{Cr}} - Q_{pBdOffset}_c \quad (8-954)$$

Téngase en cuenta que las variables $Q_{p'_{cbCr}}$, $Q_{p'_{cb}}$, $Q_{p'_{Cr}}$ se obtienen en la cláusula “8.7.1 **Proceso de derivación para los parámetros de cuantificación**”.

45 En una implementación ilustrativa, $Q_{pBdOffset}$ se resta del valor de Q_P de croma (tal como se muestra en 8-952, 8-953, 8-954) y se aplica el valor promedio (tal como se muestra en la sección revisada **8.8.3.6.3** que se presenta más adelante). En un diseño alternativo, el $Q_{pBdOffset}$ puede restarse del valor de Q_P de croma durante la etapa de cálculo del promedio. Los efectos de estos dos diseños alternativos son idénticos.

50 Los detalles de un proceso de derivación para los parámetros de cuantificación (tal como se muestra en S1611 o S1621 de la Figura 17) según algunas realizaciones de la presente descripción se describen en la sección 8.7.1 del siguiente modo:

8.7.1 Proceso de derivación de los parámetros de cuantificación

55 Las entradas de este proceso son:

- una ubicación de luma (x_{Cb} , y_{Cb}) que especifica la muestra de luma superior izquierda del bloque de codificación actual en relación con la muestra de luma superior izquierda de la imagen actual,
- 60 – una variable $cbWidth$ que especifica la anchura del bloque de codificación actual en muestras de luma,
- una variable $cbHeight$ que especifica la altura del bloque de codificación actual en muestras de luma,

65

– una variable `treeType` que especifica si se usa un árbol único (`SINGLE_TREE`) o un árbol doble para particionar las CTU y, cuando se usa un árbol doble, si los componentes de luma (`DUAL_TREE_LUMA`) o de croma (`DUAL_TREE_CHROMA`) están actualmente procesados.

5 En este proceso, se derivan el parámetro de cuantificación de luma Qp'_Y y los parámetros de cuantificación de croma Qp'_{Cb} y Qp'_{Cr} .

La ubicación de luma (xQg , yQg) especifica la muestra de luma superior izquierda del grupo de cuantificación actual en relación con la muestra de luma superior izquierda de la imagen actual. Las posiciones horizontal y vertical xQg e yQg se establecen como iguales a $CuQgTopLeftX$ y $CuQgTopLeftY$, respectivamente.

Nota: El grupo de cuantificación actual es una región rectangular dentro de un bloque de árbol de codificación que comparte el mismo qP_{Y_PRED} . Su anchura y altura son iguales a la anchura y altura del nodo del árbol de codificación cuya posición de muestra de luma superior izquierda se asigna a las variables $CuQgTopLeftX$ y $CuQgTopLeftY$.

15 Cuando `treeType` es igual a `SINGLE_TREE` o `DUAL_TREE_LUMA`, el parámetro de cuantificación de luma predicho qP_{Y_PRED} se obtiene mediante las siguientes etapas ordenadas:

1. La variable qP_{Y_PREV} se deriva del siguiente modo:
 - Si una o más de las siguientes condiciones son verdaderas, qP_{Y_PREV} se establece igual a $SliceQp_Y$:
 - El grupo de cuantificación actual es el primer grupo de cuantificación de un segmento.
 - El grupo de cuantificación actual es el primer grupo de cuantificación en un ladrillo.
 - El grupo de cuantificación actual es el primer grupo de cuantificación de una fila de CTB de un ladrillo y `entropy_coding_sync_enabled_flag` es igual a 1.

30 - De lo contrario, qP_{Y_PREV} se establece igual al parámetro de cuantificación de luma Qp_Y de la última unidad de codificación de luma del grupo de cuantificación anterior en el orden de decodificación.

2. El proceso de derivación para la disponibilidad en bloque vecinos, tal como se especifica en la cláusula 6.4.4, se invoca con la ubicación ($xCurr$, $yCurr$) establecida igual a (xCb , yCb), la ubicación vecina ($xNbY$, $yNbY$) establecida igual a ($xQg - 1$, yQg), `checkPredModeY` establecido igual a FALSO y `cldx` establecido igual a 0 como entradas, y la salida se asigna a `availableA`. La variable qP_{Y_A} se deriva del siguiente modo:

- Si una o más de las siguientes condiciones son verdaderas, qP_{Y_A} se establece igual a qP_{Y_PREV} :
- `availableA` es igual a FALSO.
- El CTB que contiene el bloque de codificación de luma que cubre la ubicación de luma ($xQg - 1$, yQg) no es igual al CTB que contiene el bloque de codificación de luma actual en (xCb , yCb), es decir, se cumplen todas las condiciones siguientes:
 - $(xQg - 1) \gg CtbLog2SizeY$ no es igual a $(xCb) \gg CtbLog2SizeY$
 - $(yQg) \gg CtbLog2SizeY$ no es igual a $(yCb) \gg CtbLog2SizeY$

50 - De lo contrario, qP_{Y_A} se establece igual al parámetro de cuantificación de luma Qp_Y de la unidad de codificación que contiene la cobertura del bloque de codificación de luma ($xQg - 1$, yQg).

3. El proceso de derivación para la disponibilidad en bloque vecinos, tal como se especifica en la cláusula 6.4.4, se invoca con la ubicación ($xCurr$, $yCurr$) establecida igual a (xCb , yCb), la ubicación vecina ($xNbY$, $yNbY$) establecida igual a (xQg , $yQg - 1$), `checkPredModeY` establecido igual a FALSO y `cldx` establecido igual a 0 como entradas, y la salida se asigna a `availableB`. La variable qP_{Y_B} se deriva del siguiente modo:

- Si una o más de las siguientes condiciones son verdaderas, qP_{Y_B} se establece igual a qP_{Y_REV} :
- `availableB` es igual a FALSO.
- El CTB que contiene el bloque de codificación de luma que cubre la ubicación de luma (xQg , $yQg - 1$) no es igual al CTB que contiene el bloque de codificación de luma actual en (xCb , yCb), es decir, se cumplen todas las condiciones siguientes:
 - $(xQg) \gg CtbLog2SizeY$ no es igual a $(xCb) \gg CtbLog2SizeY$

- $(yQg - 1) \gg \text{CtbLog2SizeY}$ no es igual a $(yCb) \gg \text{CtbLog2SizeY}$

5 - De lo contrario, qP_{Y_B} se establece igual al parámetro de cuantificación de luma Qp_Y de la unidad de codificación que contiene la cobertura del bloque de codificación de luma $(xQg, yQg - 1)$.

4. El parámetro de cuantificación de luma predicho qP_{Y_PRED} se obtiene del siguiente modo:

10 - Si se cumplen todas las condiciones siguientes, entonces qP_{Y_RED} se establece igual al parámetro de cuantificación de luma Qp_Y de la unidad de codificación que contiene la cobertura del bloque de codificación de luma $(xQg, yQg - 1)$:

- availableB es igual a VERDADERO.

15 - el grupo de cuantificación actual es el primer grupo de cuantificación de una fila de CTB dentro de un ladrillo

- De lo contrario, qP_{Y_PRED} se obtiene del siguiente modo:

$$qP_{Y_PRED} = (qP_{Y_A} + qP_{Y_B} + 1) \gg 1 \quad (8-932)$$

20 La variable Qp_Y se deriva del siguiente modo:

$$Qp_Y = ((qP_{Y_PRED} + \text{CuQpDeltaVal} + 64 + 2 * \text{QpBdOffset}_Y) \% (64 + \text{QpBdOffset}_Y)) - \text{QpBdOffset}_Y \quad (8-933)$$

25 El parámetro de cuantificación de luma Qp'_Y se obtiene del siguiente modo:

$$Qp'_Y = Qp_Y + \text{QpBdOffset}_Y \quad (8-934)$$

30 Cuando ChromaArrayType no es igual a 0 y treeType es igual a SINGLE_TREE o DUAL_TREE_CHROMA, se aplica lo siguiente:

- Cuando treeType es igual a DUAL_TREE_CHROMA, la variable Qp_Y se establece igual al parámetro de cuantificación Qp_Y de la unidad de codificación de luma que cubre la ubicación de luma $(xCb + \text{cbWidth} / 2, yCb + \text{cbHeight} / 2)$.

35 - Las variables qP_{Cb} , qP_{Cr} y qP_{CbCr} se derivan del siguiente modo:

$$qP_{\text{Chroma}} = \text{Clip3}(-\text{QpBdOffset}_C, 63, Qp_Y) \quad (8-935)$$

$$40 \quad qP_{Cb} = \text{ChromaQpTable}[0][qP_{\text{Chroma}}] \quad (8-936)$$

$$qP_{Cr} = \text{ChromaQpTable}[1][qP_{\text{Chroma}}] \quad (8-937)$$

$$qP_{CbCr} = \text{ChromaQpTable}[2][qP_{\text{Chroma}}] \quad (8-938)$$

45 - Los parámetros de cuantificación de croma para los componentes Cb y Cr, Qp'_{Cb} y Qp'_{Cr} , y la codificación conjunta Cb-Cr Qp'_{CbCr} se derivan del siguiente modo:

$$50 \quad Qp'_{Cb} = \text{Clip3}(-\text{QpBdOffset}_C, 63, qP_{Cb} + \text{pps}_{cb_qp_offset} + \text{slice}_{cb_qp_offset} + \text{CuQpOffset}_{Cb}) + \text{QpBdOffset}_C \quad (8-939)$$

$$Qp'_{Cr} = \text{Clip3}(-\text{QpBdOffset}_C, 63, qP_{Cr} + \text{pps}_{cr_qp_offset} + \text{slice}_{cr_qp_offset} + \text{CuQpOffset}_{Cr}) + \text{QpBdOffset}_C \quad (8-940)$$

$$55 \quad Qp'_{CbCr} = \text{Clip3}(-\text{QpBdOffset}_C, 63, qP_{CbCr} + \text{pps}_{cbcr_qp_offset} + \text{slice}_{cbcr_qp_offset} + \text{CuQpOffset}_{CbCr}) + \text{QpBdOffset}_C \quad (8-941)$$

60 Cabe señalar que, tal como se ilustra en 8-935, el QP de croma (p. ej., Qp_Y) no se usa directamente para derivar el QP de croma (p. ej., qP_{Cb} , qP_{Cr} o qP_{CbCr}). En un ejemplo, se puede aplicar una etapa adicional, tal como el recorte, al QP de luma (Qp_Y).

El proceso de decisión para los bordes del bloque de croma (tal como se muestra en S1601 de la Figura 16) según la solución 4 de la presente descripción se describe en la sección revisada 8.8.3.6.3 del siguiente modo:

65 8.8.3.6.3 Proceso de decisión para bordes de bloques de croma

Las variables Q_{pQ} y Q_{pP} se establecen como iguales a $Q_{p'_{CbCr}} - Q_{pBdOffsetc}$ cuando $TuCResMode[xCb][yCb]$ es igual a 2, $Q_{p'_{Cb}} - Q_{pBdOffsetc}$ cuando $cldx$ es igual a 1; $Q_{p'_{Cr}} - Q_{pBdOffsetc}$ cuando $cldx$ es igual a 2 de las unidades de codificación que incluyen los bloques de codificación que contienen la muestra $q_{0,0}$ y $p_{0,0}$, respectivamente.

5 La variable Q_{pc} se deriva del siguiente modo:

$$Q_{pc} = (Q_{pQ} + Q_{pP} + 1) \gg 1$$

10 Con referencia a la sección 8.8.3.6.3, en una implementación ilustrativa, $Q_{pBdoffset}$ se resta del valor de Q_{p} de croma (tal como se ilustra en 8-952, 8-953, 8-954) y, a continuación, el valor promedio se aplica directamente (tal como se ilustra en la sección 8.8.3.6.3). Se puede entender que la forma es sustancialmente la misma en que se resta $Q_{pBdoffset}$ durante la etapa de cálculo del promedio. Los detalles de ejemplo de la etapa 1631 (tal como se muestra en la Figura 17) se describen en la sección 8.8.3.6.3 anterior de la especificación de la VVC, y no se repetirán a continuación.

15 Solución 5:

La solución 5 es básicamente similar a la solución 4, excepto que las reglas de conversión son ligeramente diferentes. El texto de la especificación se adjunta a continuación.

20 **8.8.3.6.3 Proceso de decisión para bordes de bloques de croma**

Este proceso solo se invoca cuando $ChromaArrayType$ no es igual a 0.

Las entradas de este proceso son:

- 25 – una matriz de muestra de imagen de croma $recPicture$,
- una ubicación de croma (xCb , yCb) que especifica la muestra superior izquierda del bloque de codificación de croma actual en relación con la muestra de croma superior izquierda de la imagen actual,
- 30 – una ubicación de croma (xBl , yBl) que especifica la muestra superior izquierda del bloque de croma actual en relación con la muestra superior izquierda del bloque de codificación de croma actual,
- 35 – una variable $edgeType$ que especifica si se filtra un borde vertical ($EDGE_VER$) u horizontal ($EDGE_HOR$),
- una variable $cldx$ que especifica el índice del componente de color,
- una variable $cQpPicOffset$ que especifica el desplazamiento del parámetro de cuantificación de croma a nivel de imagen,
- 40 – una variable bS que especifica la intensidad de filtrado de límites,
- una variable $maxFilterLengthCbCr$.

45 Las salidas de este proceso son

- la variable modificada $maxFilterLengthCbCr$,
- 50 – la variable tc .

La variable $maxK$ se deriva del siguiente modo:

- Si $edgeType$ es igual a $EDGE_VER$, se aplica lo siguiente:

55 $maxK = (SubHeightC == 1) ? 3 : 1$ (8-1124)

- De lo contrario ($edgeType$ es igual a $EDGE_HOR$), se aplica lo siguiente:

60 $maxK = (SubWidthC == 1) ? 3 : 1$ (8-1125)

Los valores p_i y q_i con $i = 0..maxFilterLengthCbCr$ y $k = 0..maxK$ se derivan del siguiente modo:

- Si $edgeType$ es igual a $EDGE_VER$, se aplica lo siguiente:

65 $q_{i,k} = recPicture[xCb + xB1 + i][yCb + yB1 + k]$ (8-1126)

$$p_{i,k} = \text{recPicture}[x_{Cb} + x_{B1} - i - 1][y_{Cb} + y_{B1} + k] \quad (8-1127)$$

$$\text{subSampleC} = \text{SubHeightC} \quad (8-1128)$$

la variable Q_{pQ} se obtiene invocando el proceso de derivación de QP de croma, tal como se especifica en la cláusula 8.8.3.6.10, con la ubicación de croma $(x_{Cb} + x_{B1} + i, y_{Cb} + y_{B1} + k)$ y cldx como entradas

la variable Q_{pP} se obtiene invocando el proceso de derivación de QP de croma, tal como se especifica en la cláusula 8.8.3.6.10, con la ubicación de croma $(x_{Cb} + x_{B1} - i - 1, y_{Cb} + y_{B1} + k)$ y cldx como entradas

– De lo contrario (edgeType es igual a EDGE_HOR), se aplica lo siguiente:

$$q_{i,k} = \text{recPicture}[x_{Cb} + x_{B1} + k][y_{Cb} + y_{B1} + i] \quad (8-1129)$$

$$p_{i,k} = \text{recPicture}[x_{Cb} + x_{B1} + k][y_{Cb} + y_{B1} - i - 1] \quad (8-1130)$$

$$\text{subSampleC} = \text{SubWidthC} \quad (8-1131)$$

la variable Q_{pQ} se obtiene invocando el proceso de derivación de QP de croma, tal como se especifica en la cláusula 8.8.3.6.10, con la ubicación de croma $(x_{Cb} + x_{B1} + k, y_{Cb} + y_{B1} + i)$ y cldx como entradas

la variable Q_{pP} se obtiene invocando el proceso de derivación de QP de croma, tal como se especifica en la cláusula 8.8.3.6.10, con la ubicación de croma $(x_{Cb} + x_{B1} + k, y_{Cb} + y_{B1} - i - 1)$ y cldx como entradas

La variable Q_{pC} se deriva del siguiente modo:

$$q_{pC} = (Q_{pQ} + Q_{pP} + 1) \gg 1$$

8.8.3.6.10 Proceso de derivación de parámetros de cuantificación para bloques de codificación de croma

Este proceso solo se invoca cuando ChromaArrayType no es igual a 0.

Las entradas de este proceso son:

– Un bloque de codificación de croma que contiene una ubicación de muestra de croma dada (x_{Cb}, y_{Cb})

– una variable cldx que especifica el índice del componente de color del bloque de codificación de croma dado. La salida de este proceso es un parámetro de cuantificación q_P del bloque de codificación que contiene la muestra (x_{Cb}, y_{Cb})

– Si $\text{TuCResMode}[x_{Cb}][y_{Cb}]$ es igual a 2, se aplica lo siguiente:

$$q_P = Q_{pCbCr} \quad (8-952)$$

cQpPicOffset se establece igual a $\text{pps_joint_cbcr_qp_offset}$

– De lo contrario, si cldx es igual a 1, se aplica lo siguiente:

$$q_P = Q_{pCb} \quad (8-953)$$

– De lo contrario (cldx es igual a 2), se aplica lo siguiente:

$$q_P = Q_{pCr} \quad (8-954)$$

$$q_P = \text{Clip3}(0, 63, q_P + \text{cQpPicOffset})$$

NOTA: la variable cQpPicOffset proporciona un ajuste para el valor de pps_cb_qp_offset o pps_cr_qp_offset , según si el componente de croma filtrado es el componente de Cb o Cr. Sin embargo, para evitar la necesidad de variar la cantidad del ajuste dentro de la imagen, el proceso de filtrado no incluye un ajuste para el valor de $\text{slice_cb_qp_offset}$ o $\text{slice_cr_qp_offset}$ ni (cuando $\text{cu_chroma_qp_offset_enabled_flag}$ es igual a 1) para el valor de CuQpOffsetCb , CuQpOffsetCr , o CuQpOffsetCbCr .

NOTA: las variables Q_{pCbCr} , Q_{pCb} , Q_{pCr} se derivan como en la sección 8.7.1 (Proceso de derivación para los parámetros de cuantificación)

En la ecuación 8-935, Q_p representa un primer QP de luma (tal como Q_{pVF}) de un primer bloque de luma (801) del primer bloque (601, 601') de imagen, o un segundo QP de luma (tal como Q_{pVQ}) de un segundo bloque (802) de luma del segundo bloque (602, 602') de image.

5 En las ecuaciones 8-936, 8-937, 8-938, ChromaQPTable representa una tabla de mapeo de Q_p de croma (tal como ChromaQPTable).

10 $Q_{pBdOffsetc}$ representa el valor del desplazamiento del intervalo del parámetro de cuantificación de croma. El valor de $Q_{pBdOffsetc}$ se obtiene del siguiente modo: $Q_{pBdOffsetc} = 6 * \text{bit_depth_chroma_minus8}$ donde "bit_depth_chroma_minus8" es un parámetro señalizado en el conjunto de parámetros de secuencia (SPS, en inglés).

15 **pps_cb_qp_offset** y **pps_cr_qp_offset** especifican los desplazamientos del parámetro de cuantificación de luma Q_p y usado para derivar $Q_{p'cb}$ y $Q_{p'cr}$, respectivamente. Los valores de **pps_cb_qp_offset** y **pps_cr_qp_offset** estarán en el intervalo de -12 a +12, ambos inclusive. Cuando ChromaArrayType es igual a 0, **pps_cb_qp_offset** y **pps_cr_qp_offset** no se utilizan en el proceso de decodificación y los decodificadores ignorarán su valor.

20 **pps_joint_cbcr_qp_offset** especifica el desplazamiento del parámetro de cuantificación de luma Q_p y usado para derivar $Q_{p'cbCr}$. El valor de **ppsjoint_cbcr_qp_offset** estará en el intervalo de -12 a +12, ambos inclusive. Cuando ChromaArrayType es igual a 0 o **sps_joint_cbcr_enabled_flag** es igual a 0, **pps_joint_cbcr_qp_offset** no se utiliza en el proceso de decodificación y los decodificadores ignorarán su valor.

25 **slice_cb_qp_offset** especifica una diferencia que se añadirá al valor de **pps_cb_qp_offset** al determinar el valor del parámetro de cuantificación $Q_{p'cb}$. El valor de **slice_cb_qp_offset** estará en el intervalo de -12 a +12, ambos inclusive. Cuando **slice_cb_qp_offset** no está presente, se deduce que es igual a 0. El valor de **pps_cb_qp_offset** + **slice_cb_qp_offset** estará en el intervalo de -12 a +12, ambos inclusive.

30 **slice_cr_qp_offset** especifica una diferencia que se añadirá al valor de **pps_cr_qp_offset** al determinar el valor del parámetro de cuantificación $Q_{p'cr}$. El valor de **slice_cr_qp_offset** estará en el intervalo de -12 a +12, ambos inclusive. Cuando **slice_cr_qp_offset** no está presente, se deduce que es igual a 0. El valor de **pps_cr_qp_offset** + **slice_cr_qp_offset** estará en el intervalo de -12 a +12, ambos inclusive.

35 **slice_joint_cbcr_qp_offset** especifica una diferencia que se añadirá al valor de **pps_joint_cbcr_qp_offset** al determinar el valor del $Q_{p'cbCr}$. El valor de **slice_joint_cbcr_qp_offset** estará en el intervalo de -12 a +12, ambos inclusive. Cuando **slice_joint_cbcr_qp_offset** no está presente, se deduce que es igual a 0. El valor de **pps_joint_cbcr_qp_offset** + **slicejoint_cbcr_qp_offset** estará en el intervalo de -12 a +12, ambos inclusive.

40 **cu_chroma_qp_offset_flag** cuando está presente y es igual a 1, especifica que se usa una entrada en **cb_qp_offset_list** [] para determinar el valor de $CuQpOffset_{cb}$, una entrada correspondiente en **cr_qp_offset_list** [] se usa para determinar el valor de $CuQpOffset_{cr}$, y una entrada correspondiente en **joint_cbcr_qp_offset_list** [] se usa para determinar el valor de $CuQpOffset_{cbCr}$. **cu_chroma_qp_offset_flag** igual a 0 especifica que estas listas no se utilizan para determinar los valores de $CuQpOffset_{cb}$, $CuQpOffset_{cr}$, y $CuQpOffset_{cbCr}$.

45 **cu_chroma_qp_offset_idx**, cuando está presente, especifica el índice en **cb_qp_offset_list** [], **cr_qp_offset_list** [], y **joint_cbcr_qp_offset_list** [] que se usa para determinar el valor de $CuQpOffset_{cb}$, $CuQpOffset_{cr}$, y $CuQpOffset_{cbCr}$. Cuando está presente, el valor de **cu_chroma_qp_offset_idx** estará en el intervalo de 0 a **chroma_qp_offset_list_len_minus1**, ambos inclusive. Cuando no está presente, se deduce que el valor de **cu_chroma_qp_offset_idx** es igual a 0.

Cuando está presente **cu_chroma_qp_offset_flag**, se aplica lo siguiente:

- 50 – La variable **IsCuChromaQpOffsetCoded** se establece igual a 1.
- Las variables $CuQpOffset_{cb}$, $CuQpOffset_{cr}$, y $CuQpOffset_{cbCr}$ se derivan del siguiente modo:
- 55 – Si **cu_chroma_qp_offset_flag** es igual a 1, se aplica lo siguiente:
 - $CuQpOffset_{cb} = \text{cb_qp_offset_list}[\text{cu_chroma_qp_offset_idx}]$ (7-166)
 - $CuQpOffset_{cr} = \text{cr_qp_offset_list}[\text{cu_chroma_qp_offset_idx}]$ (7-167)
 - 60 $CuQpOffset_{cbCr} = \text{joint_cbcr_qp_offset_list}[\text{cu_chroma_qp_offset_idx}]$ (7-168)
- De lo contrario (**cu_chroma_qp_offset_flag** es igual a 0), $CuQpOffset_{cb}$, $CuQpOffset_{cr}$, y $CuQpOffset_{cbCr}$ se establecen como igual a 0.

65

cb_qp_offset_list[i], **cr_qp_offset_list[i]**, y **joint_cbc_r_qp_offset_list[i]**, especifica los desplazamientos usados en la derivación de Qp'_{cb} , Qp'_{cr} , y Qp'_{cbCr} , respectivamente. Los valores de **cb_qp_offset_list[i]**, **cr_qp_offset_list[i]** y **joint_cbc_r_qp_offset_list[i]** estarán en el intervalo de -12 a +12, ambos inclusive.

- 5 Especialmente, una realización del dispositivo es básicamente similar a una realización del método y, por lo tanto, se describe brevemente; para los procesos de ejecución de las funciones específicas de cada unidad, cabe remitirse a las descripciones parciales en la realización del método.

10 La Figura 10 es un diagrama en bloque que ilustra un aparato 1000 de filtro de desbloqueo ilustrativo según las técnicas descritas en esta descripción (se describirán más detalles a continuación, p. ej., basados en las Figuras 9A-9H o las Figuras 11-13 o las Figuras 16-17). El aparato 1000 de filtro de desbloqueo puede configurarse para realizar técnicas desbloqueo según varios ejemplos descritos en la presente memoria. En general, uno o ambos del filtro 220 de bucle de la Figura 2 y el filtro 320 de bucle de la Figura 3 pueden incluir componentes sustancialmente similares a los del aparato 1000 de filtro de desbloqueo. Otros dispositivos de codificación de vídeo, tales como codificadores de vídeo, decodificadores de vídeo, codificadores/decodificadores de vídeo (CODEC) y similares también pueden incluir componentes sustancialmente similares al filtro 1000 desbloqueo. El aparato 1000 de filtro de desbloqueo puede implementarse en hardware, software o firmware, o en cualquier combinación de los mismos. Cuando se implementa en software o firmware, también se puede proporcionar el hardware correspondiente (tal como uno o más procesadores o unidades de procesamiento y memoria para almacenar instrucciones para el software o firmware).

20 En el ejemplo de la Figura 6, el aparato 1000 de filtro de desbloqueo incluye la unidad 1004 de determinación del desbloqueo, las definiciones 1002 de soporte almacenadas en la memoria, la unidad 1006 de filtrado de desbloqueo, los parámetros 1008 de filtro de desbloqueo almacenados en la memoria, la unidad 1003 de ubicación de bordes y la estructura 1005 de datos de ubicaciones de bordes. Cualquiera o todos los componentes del filtro 1000 de desbloqueo pueden integrarse funcionalmente. Los componentes del filtro 1000 de desbloqueo se ilustran por separado solo con fines ilustrativos. En general, el filtro 1000 desbloqueo recibe datos para bloques decodificados, p. ej., desde un componente 114, 214 de suma que combina datos de predicción con datos residuales para los bloques. Los datos pueden incluir además una indicación de cómo se predijeron los bloques. En el ejemplo que se describe a continuación, el aparato 1000 de filtro de desbloqueo se configura para recibir datos que incluyen un bloque de vídeo decodificado asociado a un CTB (o una CTU) y un árbol cuaternario de CU para el CTB, donde el árbol cuaternario de CU describe cómo el CTB se divide en CU o CB y TU o TB.

25 El aparato 1000 de filtro de desbloqueo puede mantener la estructura 1005 de datos de ubicaciones de borde en una memoria del aparato 1000 de filtro de desbloqueo, o en una memoria externa proporcionada por un dispositivo de codificación de vídeo correspondiente. En algunos ejemplos, la unidad 1003 de ubicación de bordes puede recibir un árbol cuaternario correspondiente a un CTB que indica cómo se divide el CTB en CU o CB y TU o TB. La unidad 1003 de ubicación de bordes puede analizar entonces el árbol cuaternario de la CU para determinar los bordes entre los bloques de vídeo decodificados asociados con TU o CU en el CTB que son candidatos a el desbloqueo.

40 La estructura 1005 de datos de ubicaciones de bordes puede comprender una matriz que tiene una dimensión horizontal, una dimensión vertical y una dimensión representativa de los bordes horizontales y los bordes verticales. En general, los bordes entre los bloques de vídeo pueden producirse entre dos bloques de vídeo asociados con las CU de menor tamaño del CTB, o TU o CU. Suponiendo que el CTB tiene un tamaño de $N \times N$, y suponiendo que la CU de tamaño más pequeño del CTB es de un tamaño $M \times M$, la matriz puede comprender un tamaño de $[N/M] \times [N/M] \times 2$, donde "2" representa las dos direcciones posibles de los bordes entre las CU (horizontal y vertical). Por ejemplo, suponiendo que un CTB tiene 64×64 píxeles y una CU o TU de tamaño más pequeño de 8×8 , la matriz puede comprender $[8] \times [8] \times [2]$ entradas.

45 Cada entrada puede corresponder en general a un posible borde entre dos bloques de vídeo. De hecho, es posible que los bordes no existan en cada una de las posiciones dentro de la LCU correspondientes a cada una de las entradas de la estructura 1005 de datos de ubicaciones de bordes. En consecuencia, los valores de la estructura de datos pueden inicializarse como falsos. En general, la unidad 1003 de ubicación de bordes puede analizar el árbol cuaternario de la CU para determinar las ubicaciones de los bordes entre dos bloques de vídeo asociados con las TU o CU del CTB y establecer los valores correspondientes en la estructura 1005 de datos de ubicaciones de bordes como verdaderos.

50 En general, las entradas de la matriz pueden describir si existe un borde correspondiente en el CTB como candidato para el desbloqueo. Es decir, cuando la unidad 1003 de ubicación de bordes determina que existe un borde entre dos bloques de vídeo vecinos asociados con las TU o CU del CTB, la unidad 1003 de ubicación de bordes puede establecer un valor de la entrada correspondiente en la estructura 1005 de datos de ubicaciones de bordes para indicar que el borde existe (p. ej., como un valor de "verdadero").

55 La unidad 1004 de determinación de la eliminación del efecto en bloque generalmente determina si, para dos bloques vecinos, se debe eliminar el efecto en bloque en un borde entre los dos bloques. La unidad 1004 de determinación del desbloqueo puede determinar las ubicaciones de bordes utilizando la estructura 1005 de datos de ubicaciones de bordes. Cuando un valor de la estructura 1005 de datos de ubicaciones de bordes tiene un valor booleano, la unidad 1004 de determinación del desbloqueo puede determinar que un valor "verdadero" indica la presencia de un borde, y un valor "falso" indica que no hay ningún borde, en algunos ejemplos.

En general, la unidad 1004 de determinación del desbloqueo se configura con una o más funciones de determinación de desbloqueo. Las funciones pueden incluir una pluralidad de coeficientes aplicados a las líneas de píxeles que cruzan el borde entre los bloques. Por ejemplo, las funciones se pueden aplicar a una línea de píxeles que es perpendicular al borde, donde algunos píxeles están en uno de los dos bloques y algunos píxeles están en el otro de los dos bloques. Las definiciones 1002 de soporte definen el soporte para las funciones. En general, el “soporte” corresponde a los píxeles a los que se aplican las funciones.

La unidad 1004 de determinación de desbloqueo puede configurarse para aplicar una o más funciones de determinación de desbloqueo a uno o más conjuntos de soporte, tal como se define en las definiciones 1002 de soporte, para determinar si un borde particular entre dos bloques de datos de vídeo debe someterse al desbloqueo. La línea discontinua que se origina en la unidad 1004 de determinación de desbloqueo representa los datos de los bloques que se emiten sin filtrarse. En los casos en los que la unidad 1004 de determinación de desbloqueo determine que un borde entre dos bloques no debe filtrarse, el filtro 1000 desbloqueo puede emitir los datos de los bloques sin alterar los datos. Es decir, los datos pueden eludir la unidad 1006 de filtrado de desbloqueo. Por otro lado, cuando la unidad 1004 de determinación de desbloqueo determina que un borde debe someterse a desbloqueo, la unidad 1004 de determinación de desbloqueo puede hacer que la unidad 1006 de filtrado de desbloqueo filtre los valores de los píxeles cercanos al borde para desbloquear en el borde. La unidad 1006 de filtrado de desbloqueo recupera las definiciones de los filtros de desbloqueo de los parámetros 1008 de filtro de desbloqueo para los bordes que se someterán al desbloqueo, como lo indica la unidad 1004 de determinación de desbloqueo. En general, el filtrado de un borde usa valores de píxeles de la vecindad de un borde actual que se someterá a desbloqueo. Por lo tanto, tanto las funciones de decisión de desbloqueo como los filtros de desbloqueo pueden tener una determinada región de soporte a ambos lados de un borde. Al aplicar un filtro de desbloqueo a los píxeles que se encuentran en las proximidades de un borde, la unidad 1006 de filtrado de desbloqueo puede suavizar los valores de los píxeles de modo que se amortigüen las transiciones de alta frecuencia cerca del borde. De este modo, la aplicación de filtros de desbloqueo a los píxeles cercanos a un borde puede reducir los artefactos por el desbloqueo cerca del borde.

La Figura 16 es un diagrama en bloque que ilustra otro método ilustrativo para desbloquear según las técnicas descritas en esta descripción (se describirán más detalles a continuación, p. ej., basados en las Figuras 9A-9H, 10, 12, 13 y 17). El método 1600 de desbloqueo, para desbloquear un borde (903, 913, 923, 933, 943, 953, 963, 973) de bloque de croma entre un primer bloque (901, 911, 921, 931, 941, 951, 961, 971) de croma de un primer bloque (601, 601') de imagen y un segundo bloque (902, 912, 922, 932, 942, 952, 962, 972) de croma de un segundo bloque (602, 602') de imagen, en una codificación de imagen y/o una decodificación de imagen, en donde el método 1600 de desbloqueo comprende:

- realizar 1601 un proceso de decisión para el borde del bloque de croma, en donde, tal como se muestra en la Figura 17, el proceso de decisión comprende:

- determinar 1611 un parámetro de cuantificación de croma Qp_{c_p} para el primer bloque (901, 911, 921, 931, 941, 951, 961, 971) de croma basado en un primer bloque de QP de luma (tal como Qp_{v_p}) para un primer bloque (801) de luma del primer bloque (601, 601') de imagen y una o más tablas de mapeo de parámetros de cuantificación (QP) de croma;

- determinar 1621 un parámetro de cuantificación de croma Qp_{c_q} para el segundo bloque (902, 912, 922, 932, 942, 952, 962, 972) de croma basado en un segundo bloque de QP de luma (tal como Qp_{v_q}) para un segundo bloque (802) de luma del segundo bloque (602, 602') de imagen y una o más tablas de mapeo de QP de croma; y

- determinar 1631 un parámetro de cuantificación de croma Qp_c promediado y redondeado basado en el parámetro de cuantificación de croma Qp_{c_p} para el primer bloque (901, 911, 921, 931, 941, 951, 961, 971) de croma y el parámetro de cuantificación de croma Qp_{c_q} para el segundo bloque (902, 912, 922, 932, 942, 952, 962, 972) de croma;

- determinar 1641 un parámetro de umbral (tc) basado en el parámetro de cuantificación de croma (Qp_c) promediado y redondeado

- realizar 1603 un proceso de filtrado para el borde (903, 913, 923) de bloque de croma basado al menos en el parámetro de umbral (tc).

La Figura 11 es un diagrama en bloque que ilustra otro método ilustrativo para desbloquear según las técnicas descritas en esta descripción (se describirán más detalles a continuación, p. ej., basados en las Figuras 9, 10, 12 y 13).

Como se ilustra en la Figura 11, en la etapa 1101, se determina si el tamaño de ambos bloques perpendiculares y adyacentes a un borde dado en la dirección del filtro de desbloqueo es ≥ 8 muestras. En caso afirmativo, se lleva a cabo un proceso de decisión para el borde de bloque de croma. El proceso de decisión puede incluir la etapa 1103 y la etapa 1107. Los detalles de la etapa 1103 se proporcionarán más adelante. En la etapa 1107, se determina si se debe aplicar o no un filtro prolongado (tal como un filtro de pulsación más prolongada), el filtro de pulsación más prolongada de croma se puede aplicar basado en otras decisiones de filtro de pulsación más prolongada de croma. Si el tamaño del bloque es ≤ 8 muestras al menos para uno de los bloques, entonces se invoca el filtro débil normal que modifica un máximo de $(1 + 1)$ muestras basado en el valor de intensidad de límite del borde respectivo (en la etapa 1105); es decir, si la intensidad del límite (bS, en inglés) del borde es ≥ 1 , entonces se aplica el filtro débil $(1 + 1)$. Si el valor de bS es 0, entonces el filtro de desbloqueo no se aplica en absoluto para el borde dado.

Si el tamaño del bloque es ≥ 8 muestras para ambos bloques P y Q, entonces se calcula el QP promedio que se usa para determinar o derivar los valores de t_c y Beta. t_c y Beta son dos valores de umbral que se derivan de sus tablas respectivas y se indexan según el valor de QP promedio de los bloques P y Q.

Los valores de t_c y Beta se utilizan además en las decisiones de filtrado de desbloqueo que se describen anteriormente en la sección 8.8.3.6.3 y también en la sección 8.8.3.6.9 de JVET-O2001. Si se evalúa que todas las decisiones son verdaderas, se invoca al filtro de pulsación más prolongada para croma (3 + 3). El t_c se usa además en el proceso de filtrado de una muestra de croma que se describe anteriormente en la sección 8.8.3.6.9 de JVET-O2001.

Dado que las decisiones de desbloqueo dependen de los valores de umbral t_c y Beta y el proceso de filtrado para una muestra de croma depende de los umbrales t_c , al usar el método, el valor de QP de croma que se usa para los bloques que se codifican usando el modo Joint Cb-Cr (JCCR) se puede mapear correctamente a partir de su valor de QP de luma correspondiente para lograr decisiones correctas de desbloqueo y, por lo tanto, lograr una mejor calidad visual.

El experto en la técnica comprenderá que los “bloques” (“unidades”) de las diversas figuras (método y aparato) representan o describen las funcionalidades de las realizaciones de la invención (en lugar de necesariamente “unidades” individuales en hardware o software) y, por lo tanto, describirán por igual las funciones o características de las realizaciones del aparato y las realizaciones del método (unidad = etapa).

A continuación se ofrece una explicación de las aplicaciones del método de codificación, así como del método de decodificación, tal y como se muestran en las realizaciones mencionadas anteriormente, y de un sistema que los usa.

La Figura 14 es un diagrama en bloque que muestra un sistema 3100 de suministro de contenido para realizar un servicio de entrega de contenido. Este sistema 3100 de suministro de contenido incluye un dispositivo 3102 de captura y un dispositivo terminal 3106, y opcionalmente incluye una pantalla 3126. El dispositivo 3102 de captura se comunica con el dispositivo terminal 3106 por un enlace 3104 de comunicación. El enlace de comunicación puede incluir el canal 13 de comunicación descrito anteriormente. El enlace 3104 de comunicación incluye, pero no se limita a, un enlace WIFI, Ethernet, cableado, inalámbrico (3G/4G/5G), USB o cualquier tipo de combinación de los mismos, o tecnologías similares.

El dispositivo 3102 de captura genera datos, y puede codificarlos mediante el método de codificación que se ha mostrado en las realizaciones anteriores. Alternativamente, el dispositivo 3102 de captura puede distribuir los datos a un servidor de transmisión (no mostrado en las figuras), y el servidor codifica los datos y transmite los datos codificados al dispositivo terminal 3106. El dispositivo 3102 de captura incluye, pero no se limita a, una cámara, un teléfono inteligente o tableta, un ordenador o portátil, un sistema de videoconferencia, un PDA, un dispositivo montado en vehículo o una combinación de cualquiera de ellos, o dispositivos similares. Por ejemplo, el dispositivo 3102 de captura puede incluir el dispositivo 12 de origen que se ha descrito anteriormente. Cuando los datos incluyen vídeo, el codificador 20 de vídeo incluido en el dispositivo 3102 de captura puede realizar realmente un procesamiento de codificación de vídeo. Cuando los datos incluyen audio (es decir, voz), un codificador de audio incluido en el dispositivo 3102 de captura puede realizar realmente un procesamiento de codificación de audio. En algunos escenarios prácticos, el dispositivo 3102 de captura distribuye los datos de vídeo y de audio codificados multiplexándolos juntos. En otros escenarios prácticos, por ejemplo, en el sistema de videoconferencia, los datos de audio codificados y los datos de vídeo codificados no se multiplexan. El dispositivo 3102 de captura distribuye los datos de audio codificados y los datos de vídeo codificados al dispositivo terminal 3106 por separado.

En el sistema 3100 de suministro de contenido, el dispositivo terminal 310 recibe y reproduce los datos codificados. El dispositivo terminal 3106 podría ser un dispositivo con capacidad de recepción y de recuperación de datos, tal como un teléfono inteligente o tableta 3108, un ordenador o portátil 3110, una grabadora de vídeo en red (NRV, por sus siglas en inglés) / grabadora 3112 de vídeo digital (DRV, por sus siglas en inglés), un televisor 3114, un decodificador (STB) 3116, un sistema 3118 de videoconferencia, un sistema 3120 de videovigilancia, un asistente digital personal (PDA) 3122, un dispositivo 3124 montado en vehículo o una combinación de cualquiera de ellos, o dispositivos similares capaces de decodificar los datos codificados anteriormente mencionados. Por ejemplo, el dispositivo terminal 3106 puede incluir el dispositivo 14 de destino que se ha descrito anteriormente. Cuando los datos codificados incluyen vídeo, se da prioridad al decodificador 30 de vídeo incluido en el dispositivo terminal a la hora de realizar una decodificación de vídeo. Cuando los datos codificados incluyen audio, se da prioridad a un decodificador de audio incluido en el dispositivo terminal a la hora de realizar un procesamiento de decodificación de audio.

En el caso de un dispositivo terminal con su pantalla, por ejemplo, un teléfono inteligente o tableta 3108, un ordenador o portátil 3110, una grabadora de vídeo de red (NRV) / grabadora 3112 de vídeo digital (DRV), un televisor 3114, un asistente 3122 digital personal (PDA) o un dispositivo 3124 montado en vehículo, el dispositivo terminal puede suministrar los datos decodificados a su pantalla. En el caso de un dispositivo terminal no equipado con pantalla, por ejemplo, el STB 3116, el sistema 3118 de videoconferencia o el sistema 3120 de videovigilancia, se pone en contacto con una pantalla externa 3126 para que ésta reciba y muestre los datos decodificados.

Cuando cada dispositivo de este sistema realiza una codificación o una decodificación, se puede usar el dispositivo de codificación de imágenes o el dispositivo de decodificación de imágenes que se muestra en las realizaciones mencionadas anteriormente.

La Figura 15 es un diagrama que muestra una estructura de un ejemplo del dispositivo terminal 3106. Una vez que el dispositivo terminal 3106 recibe un flujo procedente del dispositivo 3102 de captura, la unidad 3202 de procesamiento de protocolos analiza el protocolo de transmisión del flujo. El protocolo incluye, pero no se limita a, el protocolo de transmisión en tiempo real (RTSP, por sus siglas en inglés), el protocolo de transferencia de hipertexto (HTTP, por sus siglas en inglés), el protocolo de transmisión en directo de HTTP (HLS, por sus siglas en inglés), MPEG-DASH, el protocolo de transporte en tiempo real (RTP, por sus siglas en inglés), el protocolo de mensajería en tiempo real (RTMP, por sus siglas en inglés) o cualquier tipo de combinación de los mismos, o protocolos similares.

Una vez que la unidad 3202 de procesamiento de protocolos procesa el flujo, se genera un archivo de flujo. Se da salida al archivo hacia una unidad 3204 de demultiplexación. La unidad 3204 de demultiplexación puede separar los datos multiplexados en los datos de audio codificados y los datos de vídeo codificados. Tal y como se ha descrito anteriormente, en algunos escenarios prácticos, por ejemplo, en el sistema de videoconferencia, los datos de audio codificados y los datos de vídeo codificados no se multiplexan. En esta situación, los datos codificados se transmiten al decodificador 3206 de vídeo y al decodificador 3208 de audio sin pasarse por la unidad 3204 de demultiplexación.

A través del procesamiento de demultiplexación, se generan un flujo elemental (ES, por sus siglas en inglés) de vídeo, un flujo ES de audio y, opcionalmente, un subtítulo. El decodificador 3206 de vídeo, que incluye el decodificador 30 de vídeo que se explicó en relación con las realizaciones mencionadas anteriormente, decodifica el flujo ES de vídeo mediante el método de decodificación que se muestra en las realizaciones mencionadas anteriormente para generar una trama de vídeo, y suministra estos datos a la unidad 3212 de sincronización. El decodificador 3208 de audio decodifica el flujo ES de audio para generar una trama de audio, y suministra estos datos a la unidad 3212 de sincronización. Alternativamente, la trama de vídeo puede almacenarse en una memoria intermedia (no mostrada en la Figura Y) antes de suministrarse a la unidad 3212 de sincronización. Similarmente, la trama de audio puede almacenarse en una memoria intermedia (no mostrada en la Figura Y) antes de suministrarse a la unidad 3212 de sincronización.

La unidad 3212 de sincronización sincroniza la trama de vídeo y la trama de audio y suministra el vídeo/audio a una pantalla 3214 de vídeo/audio. Por ejemplo, la unidad 3212 de sincronización sincroniza la presentación de la información de vídeo y de audio. En la sintaxis puede codificarse información usando marcas de tiempo relativas a la presentación los datos de audio y de vídeo codificados y marcas de tiempo relativas a la entrega del propio flujo de datos.

Si se incluyen un subtítulo en el flujo, el decodificador 3210 de subtítulos lo decodifica y lo sincroniza con la trama de vídeo y la trama de audio, y suministra el vídeo/audio/subtítulo a una pantalla 3216 de vídeo/audio/subtítulos.

La presente invención no está limitada al sistema mencionado anteriormente, y el dispositivo de codificación de imágenes o el dispositivo de decodificación de imágenes de las realizaciones mencionadas anteriormente se puede incorporar en otro sistema, por ejemplo, un sistema de automóvil.

Operadores matemáticos

Los operadores matemáticos utilizados en esta aplicación son similares a los utilizados en el lenguaje de programación C. Sin embargo, los resultados de las operaciones de división de números enteros y desplazamiento aritmético se definen con mayor precisión y se definen operaciones adicionales, como la exponenciación y la división con valores reales. Las convenciones de numeración y conteo generalmente comienzan desde 0, p. ej., “el primero” equivale al 0-ésimo, “el segundo” equivale al 1-ésimo, etc.

Operadores aritméticos

Los siguientes operadores aritméticos se definen del siguiente modo:

+ Adición

- Resta (como operador de dos argumentos) o negación (como operador de prefijo unario)

* Multiplicación, incluida la multiplicación de matrices

Exponenciación x^y . Especifica x elevado a y. En otros contextos, se utiliza dicha notación

para superíndices no destinados a la interpretación como exponenciación.

/División de números enteros con truncamiento del resultado hacia cero. Por ejemplo, $7/4$ y $-7/-4$ se truncan a 1 y $-7/4$ y $7/-4$ se truncan a -1.

+ Se usa para indicar la división en ecuaciones matemáticas donde no se pretende truncar ni redondear.

$\frac{x}{y}$

5 $\frac{x}{y}$ Se usa para indicar la división en ecuaciones matemáticas donde no se pretende truncar ni redondear.

$\sum_{i=x}^y f(i)$

10 $\sum_{i=x}^y f(i)$ La suma de $f(i)$ con i tomando todos los valores enteros desde x hasta y inclusive.

módulo $x \% y$. Resto de x dividido por y , definido solo para los números enteros x e y con $x \geq 0$ e $y > 0$.

Operadores lógicos

15 Los siguientes operadores lógicos se definen del siguiente modo:

$x \&\& y$ “y” lógico booleano de x e y

$x \|\| y$ “o” lógico booleano de x e y

20 $! \text{ “no”}$ lógico booleano

$x? y: z$ Si x es VERDADERO o no es igual a 0, se evalúa con el valor de y ; de lo contrario, se evalúa con el valor de z .

25 Operadores relacionales

Los siguientes operadores relacionales se definen del siguiente modo:

$>$ Mayor que

30 \geq Mayor que o igual a

$<$ Menor que

35 \leq Menor que o igual a

$==$ Igual a

$!=$ No igual a

40 Cuando se aplica un operador relacional a un elemento o variable de sintaxis al que se le ha asignado el valor “na” (no aplicable), el valor “na” se trata como un valor distinto para el elemento o la variable de sintaxis. Se considera que el valor “na” no es igual a ningún otro valor.

45 Operadores bit a bit

Los siguientes operadores bit a bit se definen del siguiente modo:

50 $\&$ “y” en términos de bit. Cuando se opera con argumentos enteros, la operación se realiza con una representación en complemento a dos del valor entero. Cuando se opera con un argumento binario que contiene menos bits que otro argumento, el argumento más corto se amplía añadiendo bits más significativos iguales a 0.

55 $|$ “o” bit a bit. Cuando se opera con argumentos enteros, la operación se realiza con una representación en complemento a dos del valor entero. Cuando se opera con un argumento binario que contiene menos bits que otro argumento, el argumento más corto se amplía añadiendo bits más significativos iguales a 0.

\wedge “exclusivo o” bit a bit. Cuando se opera con argumentos enteros, la operación se realiza con una representación en complemento a dos del valor entero. Cuando se opera con un argumento binario que contiene menos bits que otro argumento, el argumento más corto se amplía añadiendo bits más significativos iguales a 0.

60 $x \gg y$ Desplazamiento aritmético hacia la derecha de una representación entera en complemento a dos de x por y dígitos binarios. Esta función solo se define para valores enteros no negativos de y . Los bits desplazados adentro de los bits más significativos (MSB) como consecuencia del desplazamiento a la derecha tienen un valor igual al del bit MSB de x antes de la operación de desplazamiento.

65

$x \gg y$ Desplazamiento aritmético a la izquierda de una representación entera en complemento a dos de x por y dígitos binarios. Esta función solo se define para valores enteros no negativos de y . Los bits desplazados a los bits menos significativos (LSB) como resultado del desplazamiento a la izquierda tienen un valor igual a 0.

5 Operadores de asignación

Los siguientes operadores aritméticos se definen del siguiente modo:

= Operador de asignación

10 + + Incremento, es decir, $x++$ es equivalente a $x = x + 1$; cuando se usa en un índice de matriz, se evalúa como el valor de la variable antes de la operación de incremento.

15 - - Disminución, es decir, $x--$ equivale a $x = x - 1$; cuando se usa en un índice de matriz, se evalúa como el valor de la variable antes de la operación de disminución.

$+=$ Incremento según la cantidad especificada, es decir, $x += 3$ es equivalente a $x = x + 3$ y $x += (-3)$ es equivalente a $x = x + (-3)$.

20 $-=$ Disminución según la cantidad especificada, es decir, $x -= 3$ es equivalente a $x = x - 3$, y $x -= (-3)$ es equivalente a $x = x - (-3)$.

Notación de intervalo

25 La siguiente notación se usa para especificar un intervalo de valores:

$x = y..z$ x toma valores enteros que comienzan desde y hasta z , inclusive, siendo x, y, z números enteros y z mayor que y .

Funciones matemáticas

30 Se definen las siguientes funciones matemáticas:

35
$$\text{Abs}(x) = \begin{cases} x & ; \quad x \geq 0 \\ -x & ; \quad x < 0 \end{cases}$$

40 $\text{Asin}(x)$, la función trigonométrica de seno inverso, que opera con un argumento x que está en el intervalo de $-1,0$ a $1,0$, ambos inclusive, con un valor de salida en el intervalo de $-\pi + 2$ a $\pi + 2$, ambos inclusive, en unidades de radianes

$\text{Atan}(x)$, la función trigonométrica de tangente inversa, que opera con un argumento x , con un valor de salida en el intervalo de $-\pi + 2$ a $\pi + 2$, ambos inclusive, en unidades de radianes

45
$$\text{Atan2}(y, x) = \begin{cases} \text{Atan}\left(\frac{y}{x}\right) & ; \quad x > 0 \\ \text{Atan}\left(\frac{y}{x}\right) + \pi & ; \quad x < 0 \ \&\& \ y \geq 0 \\ \text{Atan}\left(\frac{y}{x}\right) - \pi & ; \quad x < 0 \ \&\& \ y < 0 \\ +\frac{\pi}{2} & ; \quad x == 0 \ \&\& \ y \geq 0 \\ -\frac{\pi}{2} & ; \quad \text{otherwise} \end{cases}$$

$\text{Ceil}(x)$, el número entero más pequeño mayor o igual a x .

60 $\text{Clip1}_v(x) = \text{Clip3}(0, (1 \ll \text{BitDepth}_v) - 1, x)$

$\text{Clip1}_c(x) = \text{Clip3}(0, (1 \ll \text{BitDepth}_c) - 1, x)$

65

5

$$\text{Clip3}(x, y, z) = \begin{cases} x & ; & z < x \\ y & ; & z > y \\ z & ; & \text{otherwise} \end{cases}$$

Cos (x), la función coseno trigonométrica que opera con un argumento x en unidades de radianes.

10

Floor (x), el entero más grande menor o igual a x.

15

$$\text{GetCurrMsb}(a, b, c, d) = \begin{cases} c + d & ; & b - a \geq d / 2 \\ c - d & ; & a - b > d / 2 \\ c & ; & \text{otherwise} \end{cases}$$

Ln (x), el logaritmo natural de x (el logaritmo en base e, donde e es la constante base del logaritmo natural 2.718 281 828...).

20

Log2 (x), el logaritmo en base 2 de x.

Log10 (x), el logaritmo en base 10 de x.

25

$$\text{Min}(x, y) = \begin{cases} x & ; & x \leq y \\ y & ; & x > y \end{cases}$$

30

$$\text{Max}(x, y) = \begin{cases} x & ; & x \geq y \\ y & ; & x < y \end{cases}$$

35

Round (x) = Sign(x) * Floor(Abs (x) + 0,5)

40

$$\text{Sign}(x) = \begin{cases} 1 & ; & x > 0 \\ 0 & ; & x == 0 \\ -1 & ; & x < 0 \end{cases}$$

Sin(x), la función seno trigonométrica que opera con un argumento x en unidades de radianes

45

$$\text{Sqrt}(x) = \sqrt{x}$$

Swap (x, y) = (y, x)

50

Tan (x), la función tangente trigonométrica que opera con un argumento x en unidades de radianes

Orden de prioridad de las operaciones

55

Cuando el orden de prioridad en una expresión no se indica explícitamente mediante el uso de paréntesis, se aplican las siguientes reglas:

–Las operaciones de mayor prioridad se evalúan antes que cualquier operación de menor prioridad.

60

–Las operaciones de la misma prioridad se evalúan secuencialmente de izquierda a derecha.

En la siguiente tabla se especifica la prioridad de las operaciones de mayor a menor; una posición más alta en la tabla indica una prioridad más alta.

65

Para los operadores que también se utilizan en el lenguaje de programación C, el orden de prioridad utilizado en esta especificación es el mismo que se utiliza en el lenguaje de programación C.

Tabla: Prioridad de operación desde la más alta (en la parte superior de la tabla) hasta la más baja (en la parte inferior de la tabla)

5	Operaciones (con operandos x, y, z)
	“x++”, “x- -”
	“!x”, “-x” (como operador de prefijo unitario)
	x^y
10	$\frac{x}{y}$ “x * y”, “x / y”, “x ÷ y”, “x % y”
	$\sum_{i=1}^x f(i)$
15	“x + y”, “x - y” (como operador de dos argumentos), “x << y”, “x >> y”
	“x < y”, “x <= y”, “x > y”, “x >= y”
20	“x == y”, “x != y”
	“x & y”
	“x y”
25	“x && y”
	“x y”
	“x ? y : z”
	“x..y”
30	“x = y”, “x += y”, “x -= y”

Descripción textual de las operaciones lógicas

35 En el texto, una declaración de operaciones lógicas se describiría matemáticamente de la siguiente forma:

if(condition 0)

statement 0

40 else if(condition 1)

statement 1

45 ...

else /* informative remark on remaining condition */

statement n

50 puede describirse del siguiente modo:

... como sigue / ... se aplica lo siguiente:

55 – Si la condición es 0, la declaración es 0

– De lo contrario, si la condición es 1, la declaración es 1

– ...

60 – De lo contrario (comentario informativo sobre la condición restante), la declaración es n

65 Cada declaración “Si... De lo contrario, si... De lo contrario, ...” es introducida en el texto con “... como sigue” o “... se aplica lo siguiente” seguida inmediatamente de “Si ...”. La última condición de las declaraciones “Si... De lo contrario, si... De lo contrario, ...” es siempre un “De lo contrario, ...”. Intercalado “Si... De lo contrario, si... De lo contrario, ...” se pueden identificar haciendo coincidir “... como sigue” o “... se aplica lo siguiente” con el final “De lo contrario, ...”.

En el texto, una declaración de operaciones lógicas se describiría matemáticamente de la siguiente forma

5 if(condition 0a && condition 0b)
 statement 0
 else if(condition 1a || condition 1b)
 10 statement 1
 ...
 else
 15 statement n

puede describirse del siguiente modo:

20 ... como sigue / ... se aplica lo siguiente:

- Si se cumplen todas las condiciones siguientes, declaración 0:
- 25 - condición 0a
- condición 0b
- De lo contrario, si se cumplen una o más de las siguientes condiciones, declaración 1:
- 30 - condición 1a
- condición 1b
- 35 - ...
- De lo contrario, la declaración es n

En el texto, una declaración de operaciones lógicas se describiría matemáticamente de la siguiente forma:

40 if(condition 0)
 statement 0
 45 si(condición 1)
 statement 1

puede describirse del siguiente modo:

- 50 Cuando la condición es 0, la declaración es 0
- Cuando la condición es 1, la declaración es 1

55 Aunque se hayan descrito realizaciones de la presente invención basadas principalmente en la codificación de vídeo, cabe señalar que las realizaciones del sistema 10 de codificación, el codificador 20 y el decodificador 30 (y, correspondientemente, el sistema 10) y las demás realizaciones descritas en la presente memoria también pueden configurarse para un procesamiento o una codificación de imágenes fijas, es decir, el procesamiento o la codificación de una imagen individual independiente de cualquier imagen anterior o consecutiva, como en la codificación de vídeo. En general, en caso de que la codificación de procesamiento de imágenes esté limitada a una sola imagen 17, puede que no estén disponibles únicamente las unidades 244 (codificador) y 344 (decodificador) de interpredicción. Todas las demás funcionalidades (denominadas también herramientas o tecnologías) del codificador 20 de vídeo y del decodificador 30 de vídeo pueden usarse igualmente para el procesamiento de imágenes fijas, p. ej., el cálculo residual 204/304, la transformada 206, la cuantificación 208, la cuantificación inversa 210/310, la transformada (inversa) 212/312, la partición 262/362, la intrapredicción 254/354 y/o el filtrado 220, 320 de bucle y la codificación entrópica 270 y la decodificación entrópica 304.

65

Las realizaciones de, p. ej. el codificador 20 y del decodificador 30, y las funciones descritas en la presente memoria con referencia a, p. ej. el codificador 20 y el decodificador 30, pueden implementarse en hardware, software, firmware o cualquier combinación de los mismos. Si se implementan en software, las funciones pueden almacenarse en un medio legible por ordenador o transmitirse a través de unos medios de comunicación como una o más instrucciones o un código y ser ejecutadas por una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir unos medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible tal como unos medios de almacenamiento de datos, o unos medios de comunicación que incluyen cualquier medio que facilite la transferencia de un programa informático de un lugar a otro, por ejemplo, según un protocolo de comunicación. De esta manera, los medios legibles por ordenador pueden corresponder en general a (1) unos medios de almacenamiento legibles por ordenador tangibles que sean no transitorios o a (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que puedan acceder uno o más ordenadores o uno o más procesadores para recuperar instrucciones, un código y/o estructuras de datos para la implementación de las tecnologías descritas en esta descripción. Un producto de programa informático puede incluir un medio legible por ordenador.

A modo de ejemplo no limitativo, tal medio de almacenamiento legible por ordenador puede comprender una RAM, una ROM, una EEPROM, un CD-ROM u otro almacenamiento de disco óptico, almacenamiento de disco magnético u otro dispositivo de almacenamiento magnético, o cualquier otro medio que pueda usarse para almacenar un código de programa deseado en forma de instrucciones o de estructuras de datos y al que pueda acceder un ordenador. Además, toda conexión se denomina apropiadamente medio legible por ordenador. Por ejemplo, si se transmiten instrucciones desde un sitio web, un servidor u otra fuente remota usando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL, por sus siglas en inglés) o tecnologías inalámbricas tales como rayos infrarrojos, ondas de radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la línea DSL o las tecnologías inalámbricas tales como los rayos infrarrojos, las ondas de radio y las microondas se incluyen en las definiciones de medio. Sin embargo, debe entenderse que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que en realidad están dirigidos a medios de almacenamiento tangibles y no transitorios. Tal y como se usan en la presente memoria, los términos disco (disk) y disco (disc) incluyen un disco compacto (CD), un disco láser, un disco óptico, un disco versátil digital (DVD), un disquete y un disco Blu-ray, donde los discos (disks) normalmente reproducen datos magnéticamente, mientras que los discos (discs) reproducen datos ópticamente con láseres. En el alcance de los medios legibles por ordenador también debe incluirse una combinación de los medios anteriores.

Las instrucciones pueden ser ejecutadas por uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de uso general, circuitos integrados de aplicación específica (ASIC), matrices de puertas lógicas programables en campo (FPGA) u otra circuitería lógica integrada o discreta equivalente. Por consiguiente, el término "procesador", tal y como se usa en la presente memoria, puede referirse a cualquiera de las anteriores estructuras o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en la presente memoria. Además, en algunos aspectos, la funcionalidad descrita en la presente memoria puede proporcionarse dentro de módulos de hardware y/o de software dedicados configurados para codificar y decodificar o incorporarse en un códec combinado. Además, las tecnologías podrían implementarse por completo en uno o más circuitos o elementos lógicos.

Las tecnologías de esta descripción pueden implementarse en una gran variedad de dispositivos o aparatos, incluidos unos cascos inalámbricos, un circuito integrado (IC, por sus siglas en inglés) o un conjunto de circuitos IC (p. ej., un conjunto de chips). En esta descripción se describen diversos componentes, módulos o unidades para recalcar aspectos funcionales de dispositivos que están configurados para realizar las técnicas descritas, pero no requieren necesariamente una realización por medio de distintas unidades de hardware. Más bien, tal y como se ha descrito anteriormente, diversas unidades pueden combinarse en una unidad de hardware de códec o proporcionarse mediante una colección de unidades de hardware interoperativas, incluyendo uno o más procesadores como los descritos anteriormente, en combinación con un software y/o un firmware apropiado(s).

REIVINDICACIONES

1. Un método de desbloqueo para desbloquear un borde de bloque de croma entre un primer bloque de croma de un primer bloque de imagen y un segundo bloque de croma de un segundo bloque de imagen en una codificación de imágenes o una decodificación de imágenes, que comprende:

realizar un proceso de decisión para el borde de bloque de croma; y

realizar un proceso de filtrado para el borde de bloque de croma al menos basado en un parámetro de umbral (t_c);

en donde el proceso de decisión comprende:

determinar un primer parámetro de cuantificación de croma (Q_{pCb}) para el primer bloque de croma, en donde se determina el primer parámetro de cuantificación de croma (Q_{pCp}) basado en un primer parámetro de cuantificación de luma (Q_{pVP}) de un primer bloque de luma del primer bloque de imagen y una tabla de mapeo de parámetros de cuantificación (QP) de croma para el primer bloque de croma, en donde la tabla de mapeo de QP de croma para el primer bloque de croma es una primera tabla de mapeo de QP de croma para un bloque codificado residual de Cb-Cr conjunto, JCCR, cuando el primer bloque de croma es el bloque codificado de JCCR del primer bloque de imagen, o la tabla de mapeo de QP de croma para el primer bloque de croma es una segunda tabla de mapeo de QP de croma para un primer componente de croma (Cb) cuando el primer bloque de croma es el primer componente de croma (Cb) del primer bloque de imagen, o la tabla de mapeo de QP de croma para el primer bloque de croma es una tercera tabla de mapeo de QP de croma para un segundo componente de croma (Cr) cuando el primer bloque de croma es el segundo componente de croma (Cr) del primer bloque de imagen;

determinar un segundo parámetro de cuantificación de croma (Q_{pCq}) para el segundo bloque de croma, en donde el segundo parámetro de cuantificación de croma (Q_{pCq}) se determina basado en un segundo parámetro de cuantificación de luma (Q_{pVQ}) de un segundo bloque de luma del segundo bloque de imagen y una tabla de mapeo de QP de croma para el segundo bloque de croma, en donde la tabla de mapeo de QP de croma para el segundo bloque de croma es una primera tabla de mapeo de QP de croma para un bloque codificado residual de Cb-Cr conjunto, JCCR, cuando el segundo bloque de croma es el bloque codificado de JCCR del segundo bloque de imagen, o la tabla de mapeo de QP de croma para el segundo bloque de croma es una segunda tabla de mapeo de QP de croma para un primer componente de croma (Cb) cuando el segundo bloque de croma es el primer componente de croma (Cb) del segundo bloque de imagen, o la tabla de mapeo de QP de croma para el segundo bloque de croma es una tercera tabla de mapeo de QP de croma para un segundo componente de croma (Cr) cuando el segundo bloque de croma es el segundo componente de croma (Cr) del segundo bloque de imagen;

determinar un tercer parámetro de cuantificación de croma promediado (Q_{pC}) basado en el primer parámetro de cuantificación de croma (Q_{pCb}) para el primer bloque de croma y el segundo parámetro de cuantificación de croma (Q_{pCq}) para el segundo bloque de croma; y

determinar el parámetro de umbral (t_c) basado en el tercer parámetro de cuantificación de croma promediado (Q_{pC});

en donde la determinación del parámetro de umbral (t_c) basado en el tercer parámetro de cuantificación de croma promediado (Q_{pC}) comprende: determinar un valor de una segunda variable de umbral (t_c') basado en el tercer parámetro de cuantificación de croma promediado (Q_{pC}) usando una tabla de mapeo; y derivar un valor del parámetro de umbral (t_c) basado en el valor de la segunda variable de umbral (t_c');

y en donde

al menos uno del primer bloque de croma y el segundo bloque de croma es un bloque codificado residual de Cb-Cr conjunto, JCCR; y la determinación del tercer parámetro de cuantificación de croma promediado (Q_{pC}) basado en el primer parámetro de cuantificación de croma (Q_{pCb}) para el primer bloque de croma y el segundo parámetro de cuantificación de croma (Q_{pCq}) para el segundo bloque de croma, comprende:

determinar el tercer parámetro de cuantificación de croma promediado (Q_{pC}) según la siguiente ecuación:

$$Q_{pC} = (Q_{pCb} + Q_{pCq} + 1) \gg 1.$$

2. El método de desbloqueo de la reivindicación 1,

en donde el primer bloque de croma es un bloque codificado residual de Cb-Cr conjunto, JCCR, del primer bloque de imagen, y el segundo bloque de croma es un bloque codificado residual de Cb-Cr conjunto (JCCR) del segundo bloque de imagen.

3. El método de desbloqueo de la reivindicación 1,

en donde el primer bloque de croma es un bloque codificado residual de Cb-Cr conjunto, JCCR, del primer bloque de imagen, y el segundo bloque de croma es un primer componente de croma del segundo bloque de imagen.

- 5 4. El método de desbloqueo de la reivindicación 1,
en donde el primer bloque de croma es un bloque codificado residual de Cb-Cr conjunto, JCCR, del primer bloque de imagen, y el segundo bloque de croma es un segundo componente de croma del segundo bloque de imagen.
- 10 5. El método de desbloqueo de la reivindicación 1,
en donde el primer bloque de croma es un primer componente de croma del primer bloque de imagen, y el segundo bloque de croma es un bloque codificado residual de Cb-Cr conjunto, JCCR, del segundo bloque de imagen.
- 15 6. El método de desbloqueo de la reivindicación 1,
en donde el primer bloque de croma es un segundo componente de croma del primer bloque de imagen, y el segundo bloque de croma es un bloque codificado residual de Cb-Cr conjunto, JCCR, del segundo bloque de imagen.
- 20 7. El método de desbloqueo de la reivindicación 1,
en donde la primera tabla de mapeo de QP de croma, la segunda tabla de mapeo de QP de croma y la tercera tabla de mapeo de QP de croma se indican o indexan mediante un primer valor de índice, un segundo valor de índice y un tercer valor de índice, respectivamente.
- 25 8. El método de desbloqueo de la reivindicación 7,
en donde el primer valor de índice es 3, el segundo valor de índice es 1 y el tercer índice es 2.
- 30 9. El método de desbloqueo de una cualquiera de las reivindicaciones anteriores,
en donde el bloque codificado de Cb-Cr conjunto se codifica usando un modo de JCCR y el modo de JCCR es un segundo modo de un conjunto de modos de JCCR disponibles.
- 35 10. El método de desbloqueo de una cualquiera de las reivindicaciones anteriores, en donde el primer bloque de imagen y el segundo bloque de imagen son bloques de transformada o el primer bloque de imagen y el segundo bloque de imagen son bloques de codificación.
- 40 11. Un aparato de filtro de desbloqueo para usar en un codificador de imágenes o un decodificador de imágenes, para desbloquear en borde de bloque de croma entre un primer bloque de croma de un primer bloque de imagen y un segundo bloque de croma de un segundo bloque de imagen, en donde el aparato de filtro de desbloqueo se configura para realizar el método de desbloqueo de una cualquiera de las reivindicaciones 1 a 10.
- 45 12. Un producto de programa informático con instrucciones de programa para realizar el método según cualquiera de las reivindicaciones 1 a 10 cuando las instrucciones de programa se ejecutan en un ordenador.

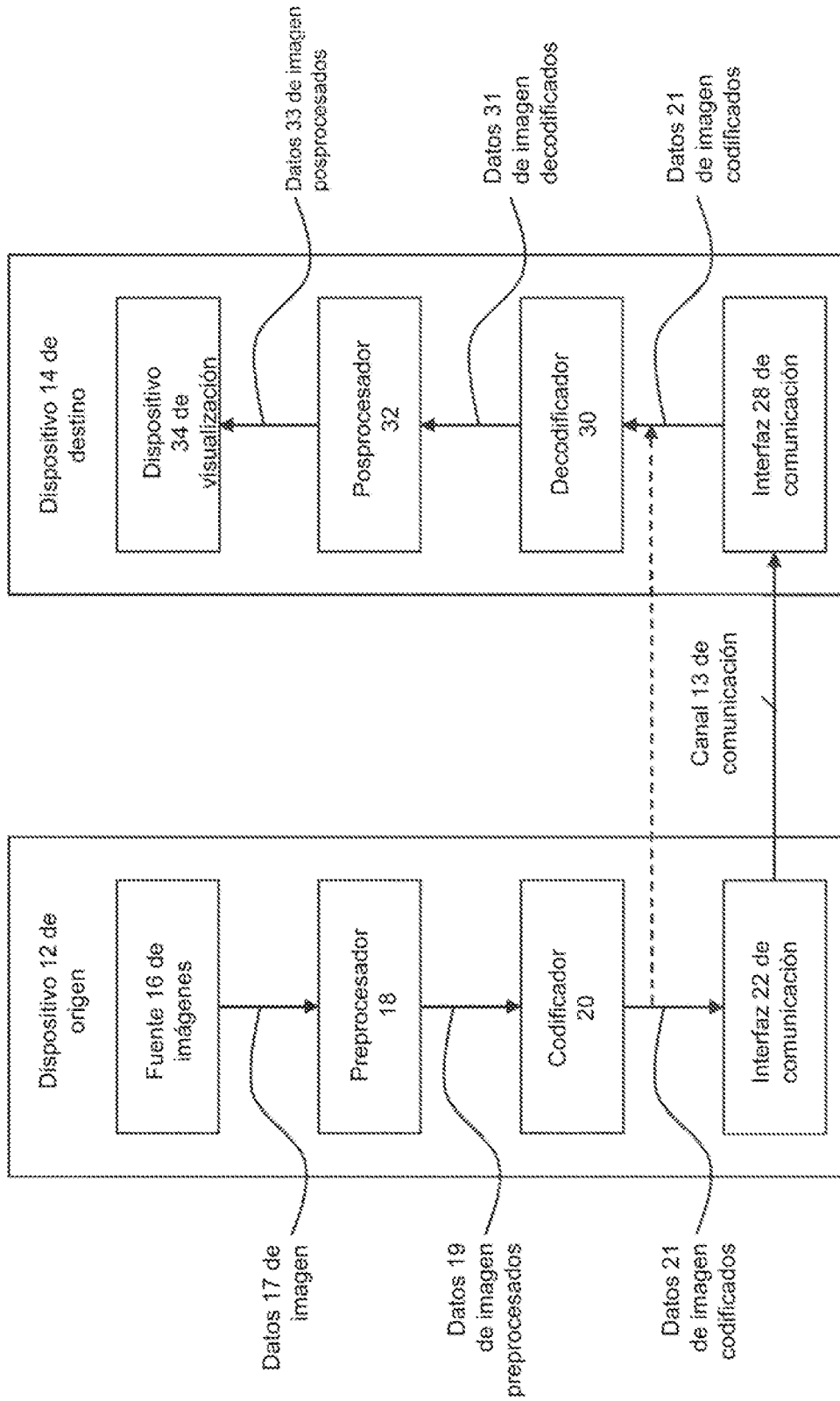


Figura 1A

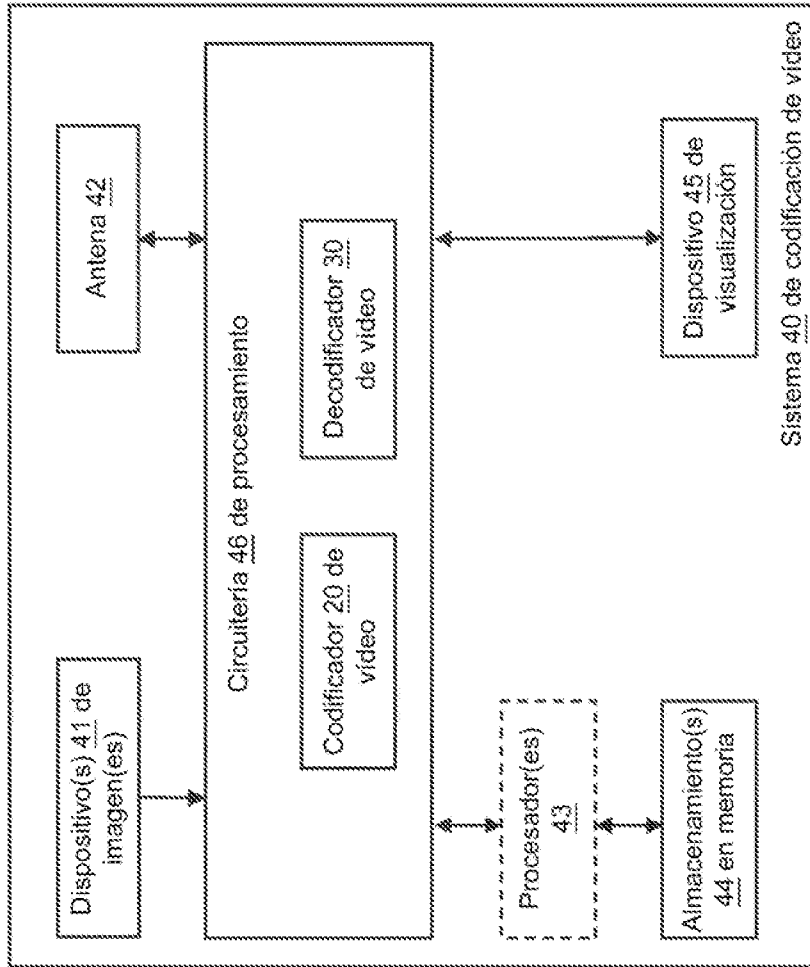


Figura 1B

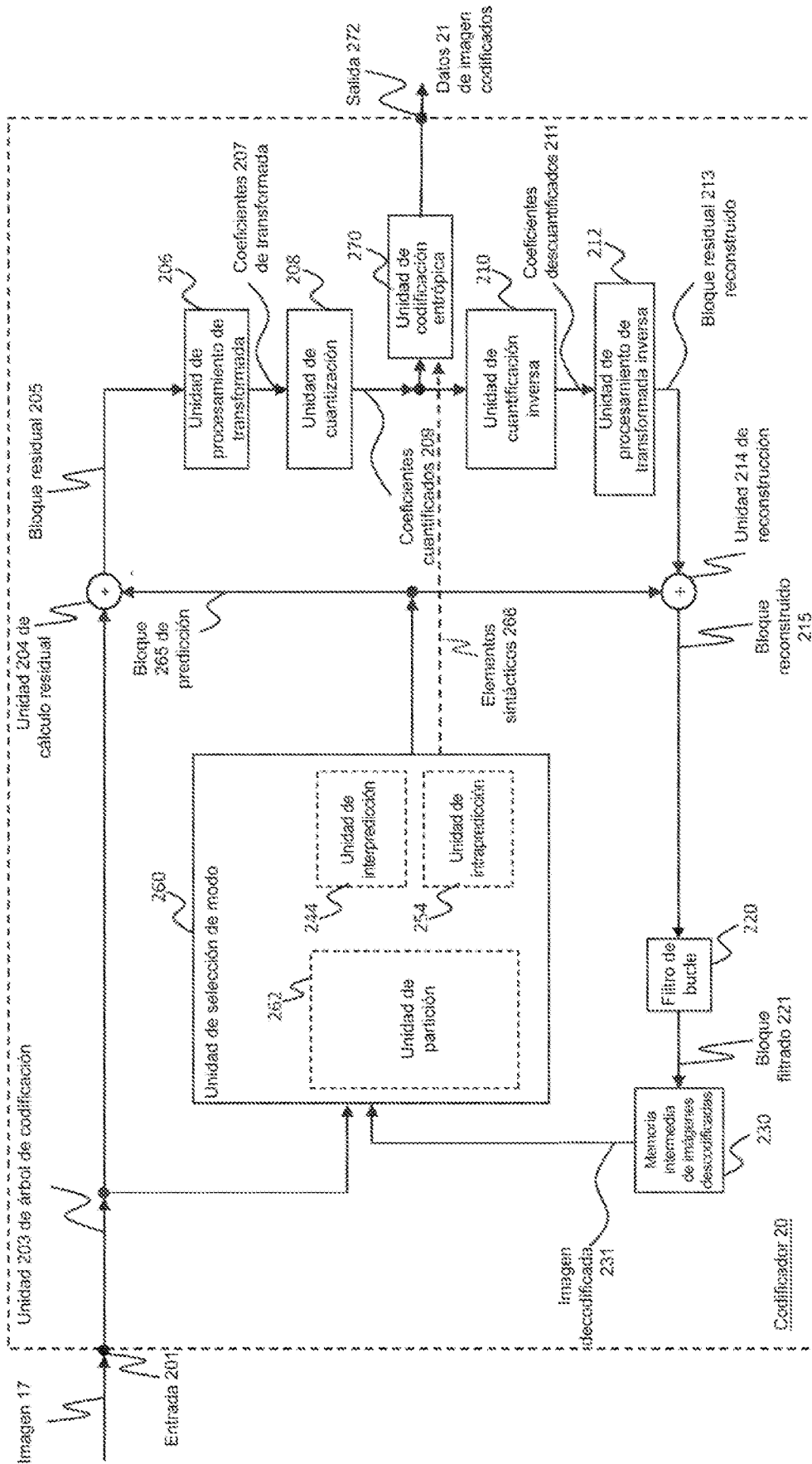


Figura 2

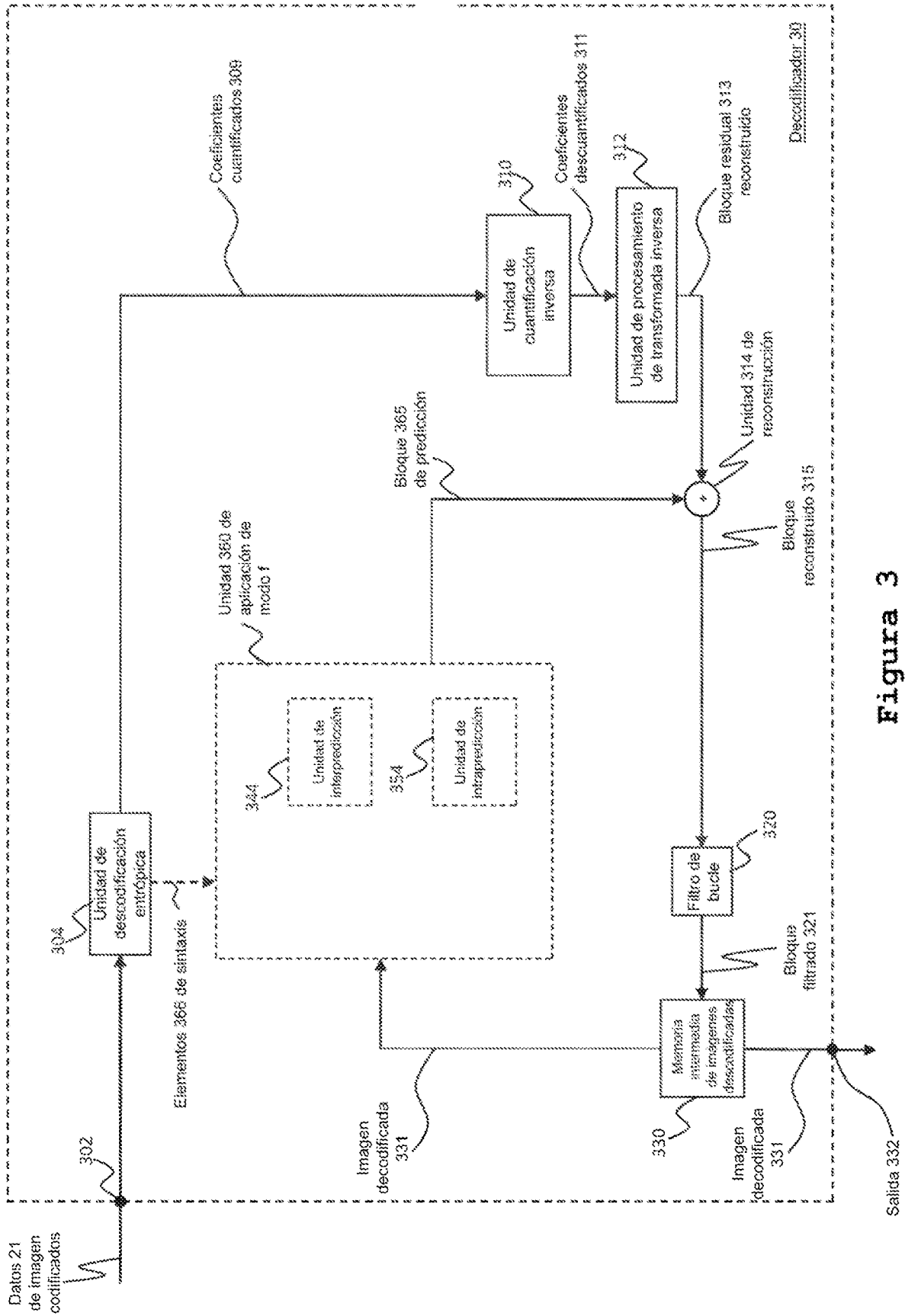


Figura 3

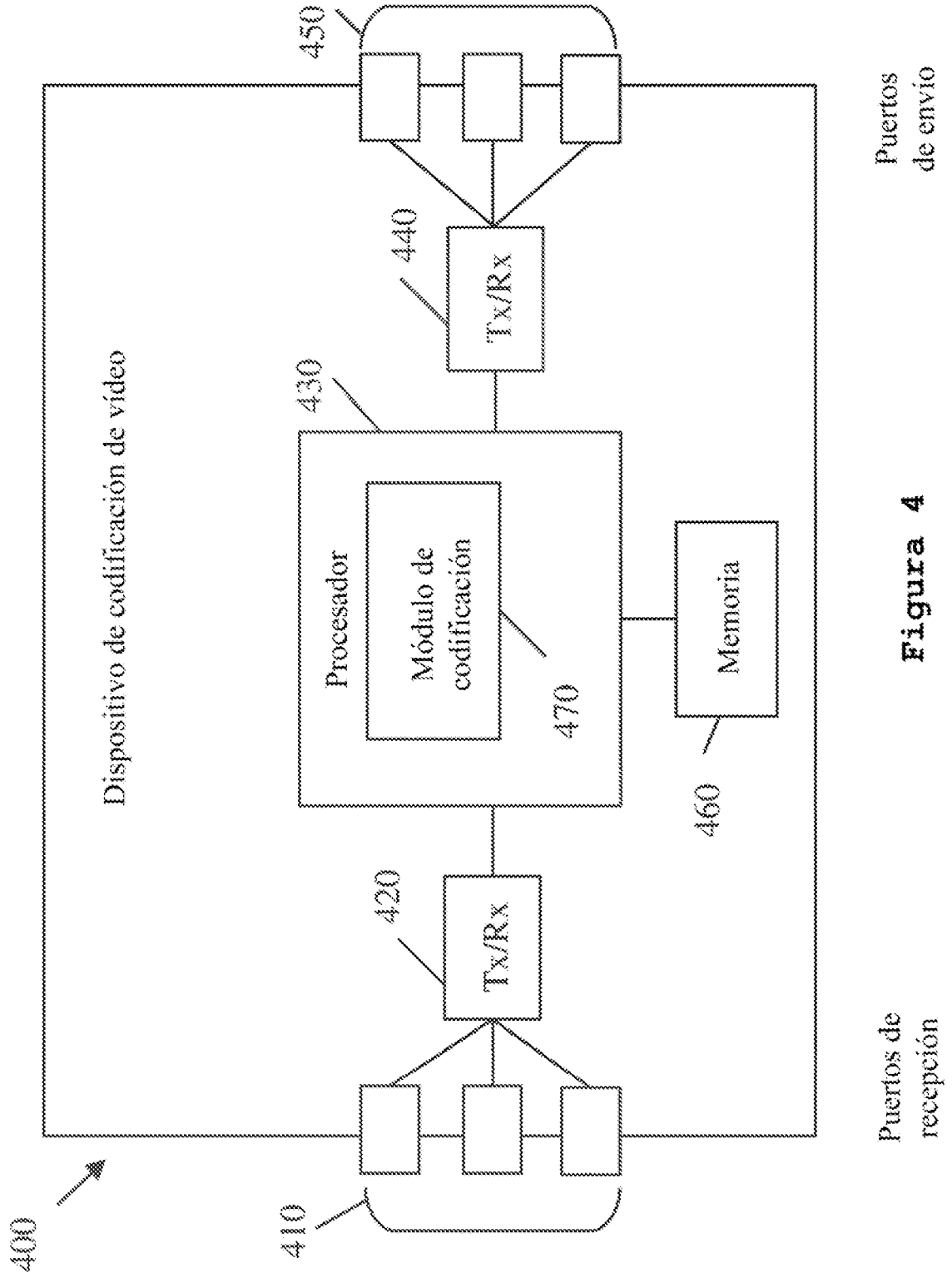


Figura 4

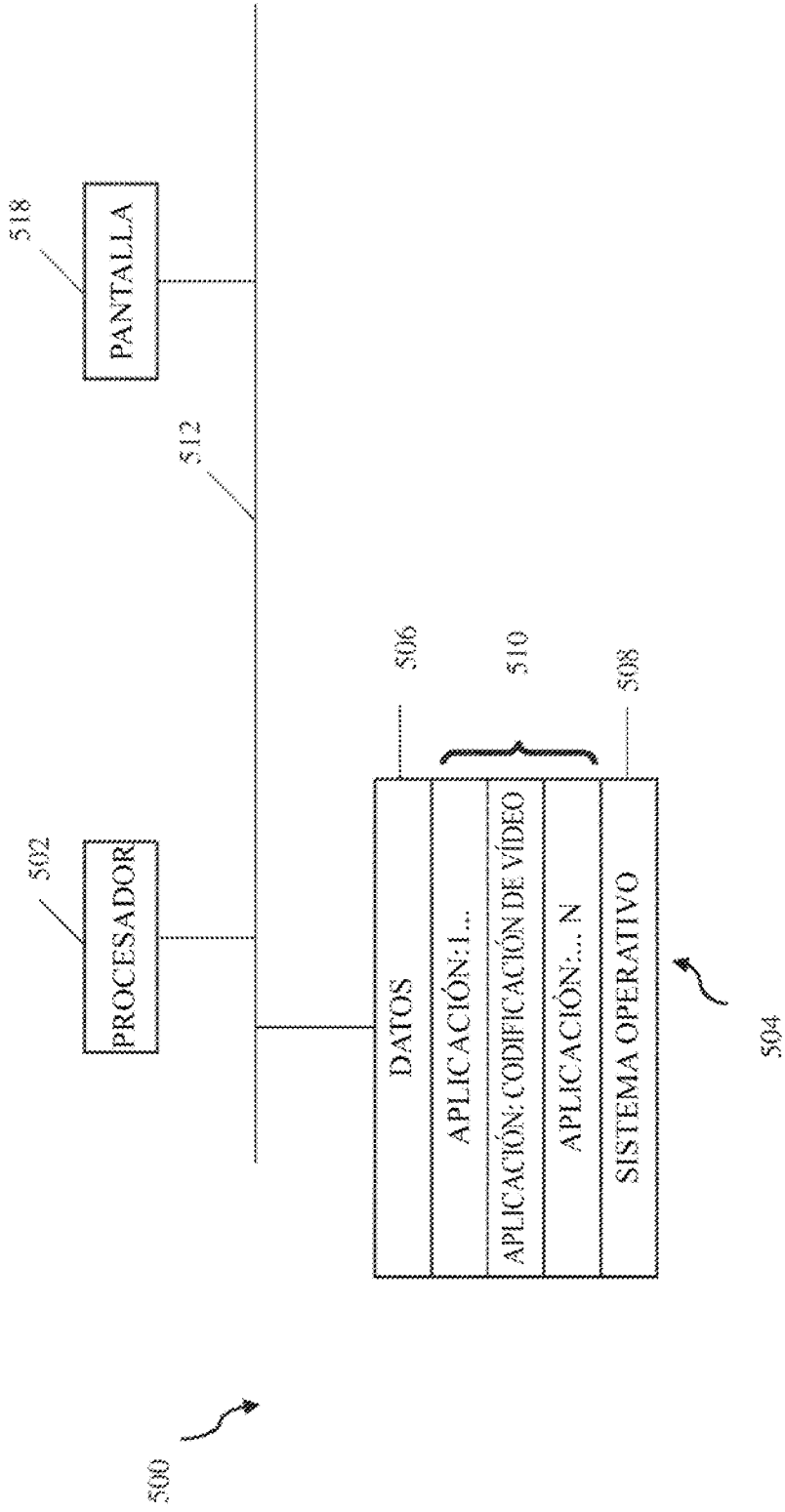


Figura 5

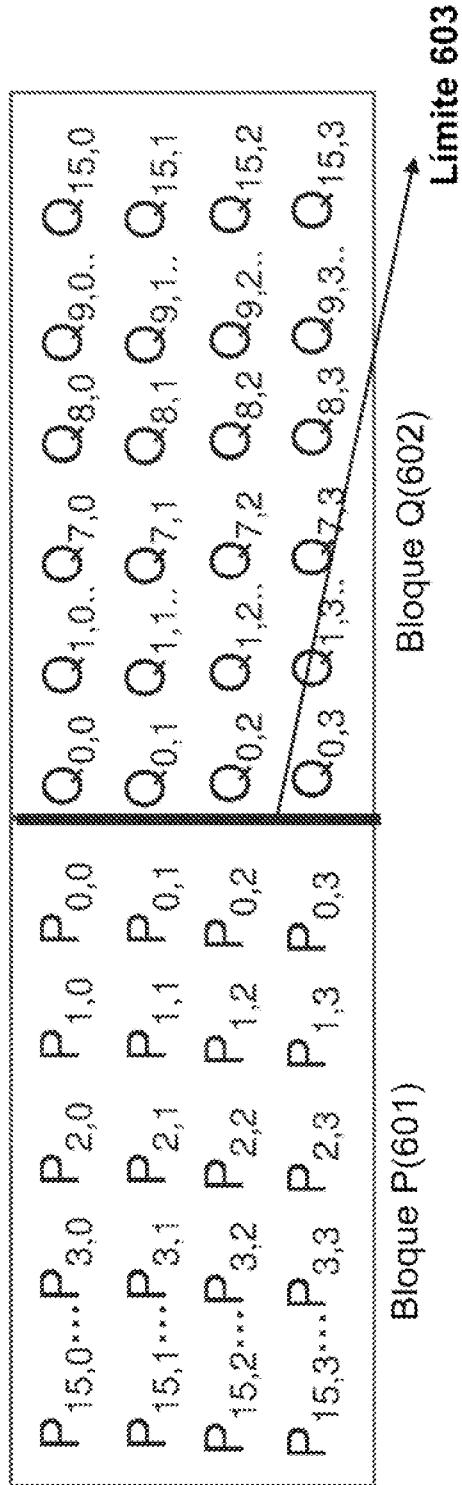


Figura 6A

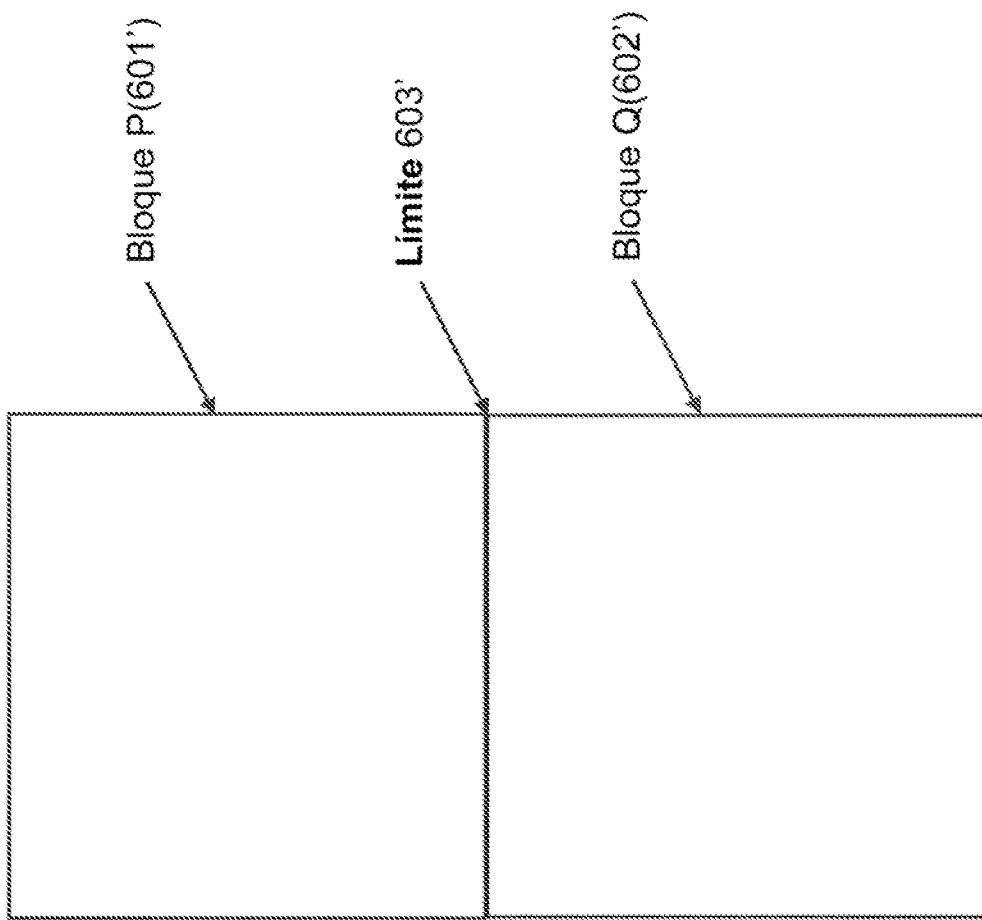


Figura 6B

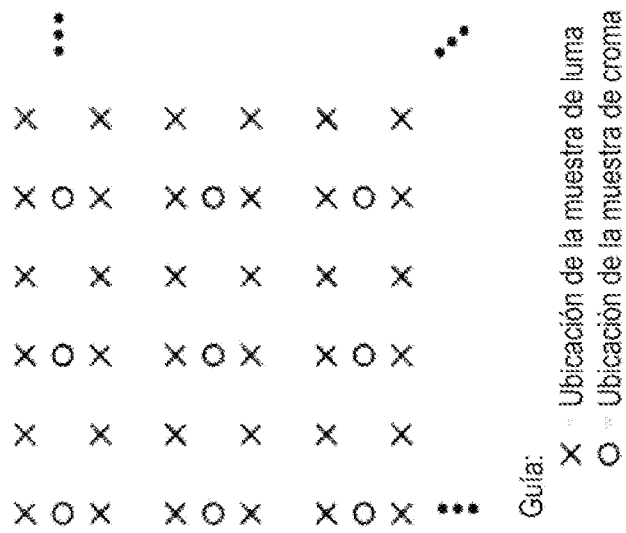


Figura 7A

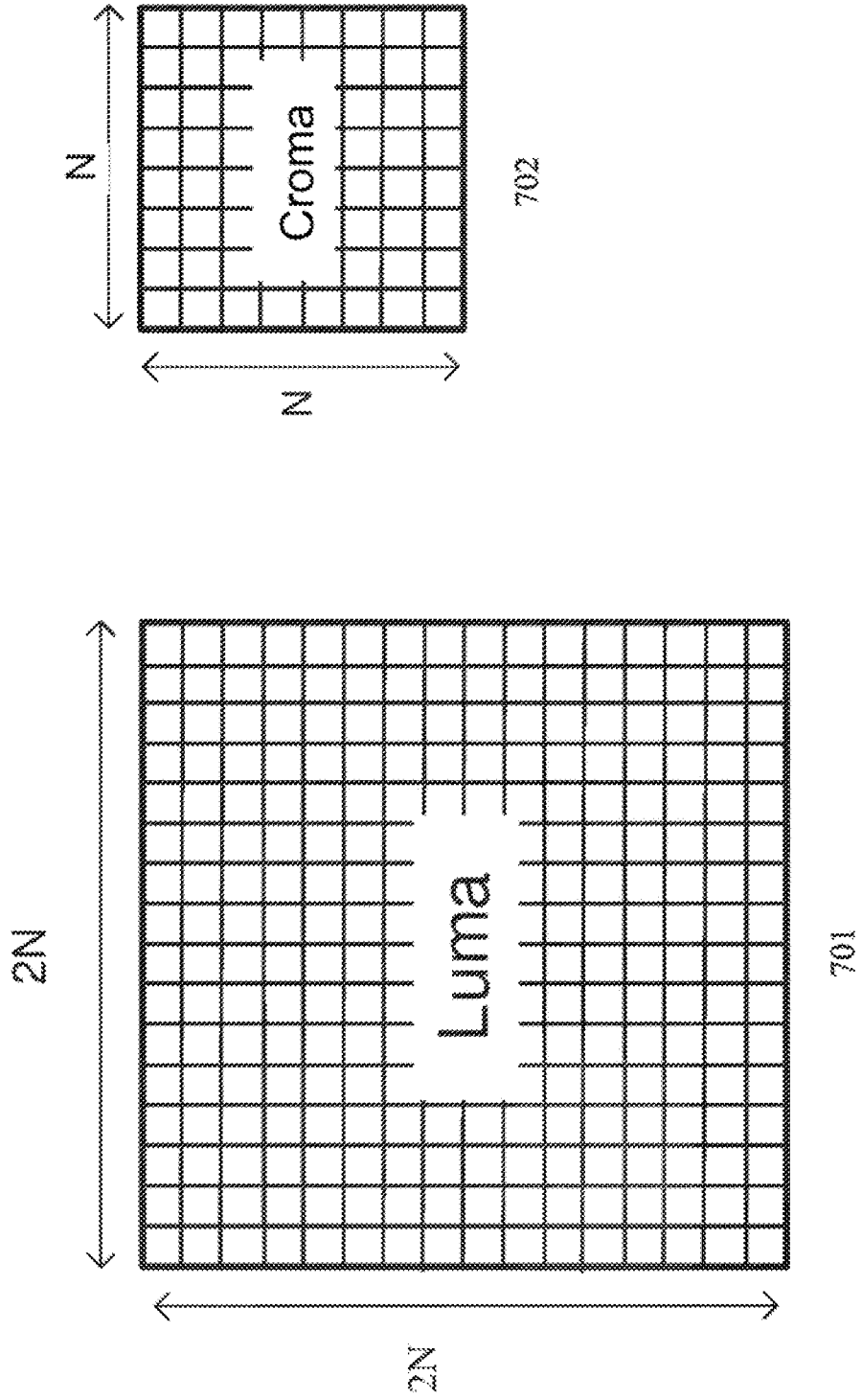


Figura 7B

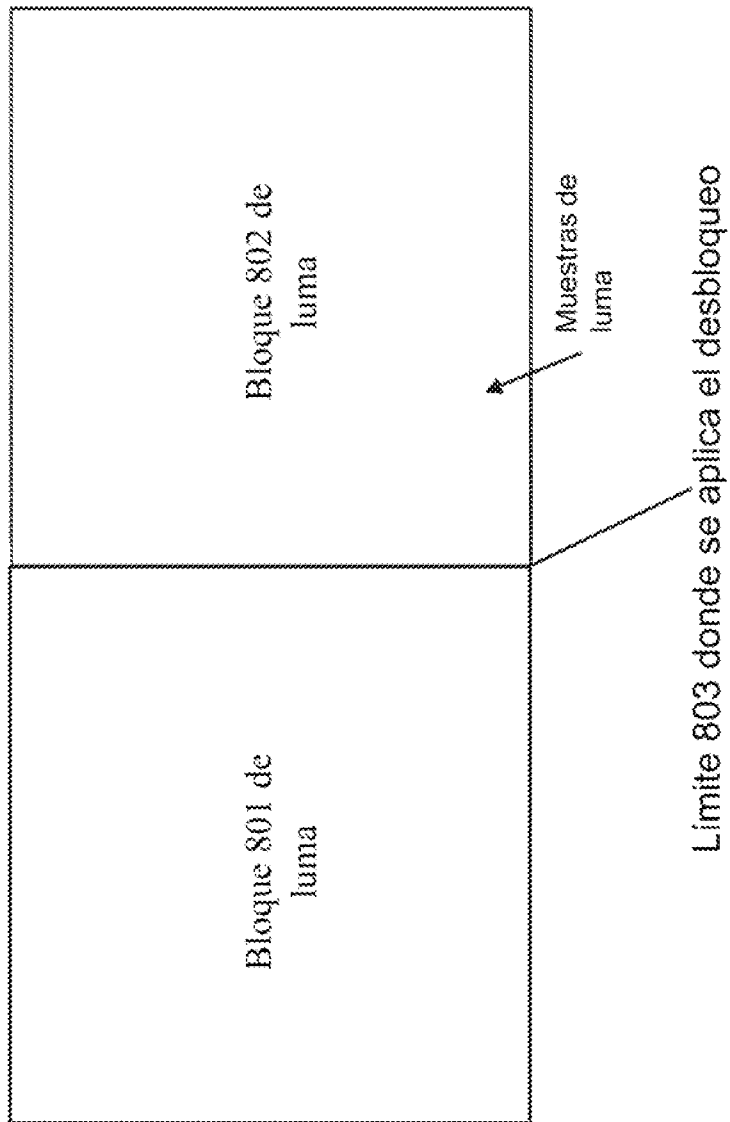
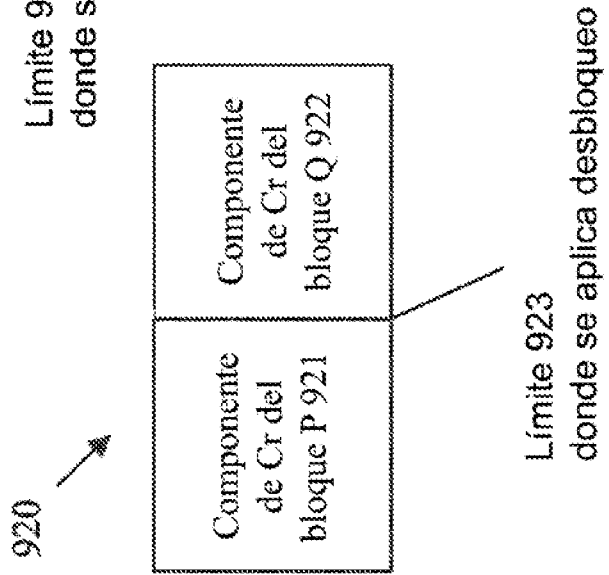
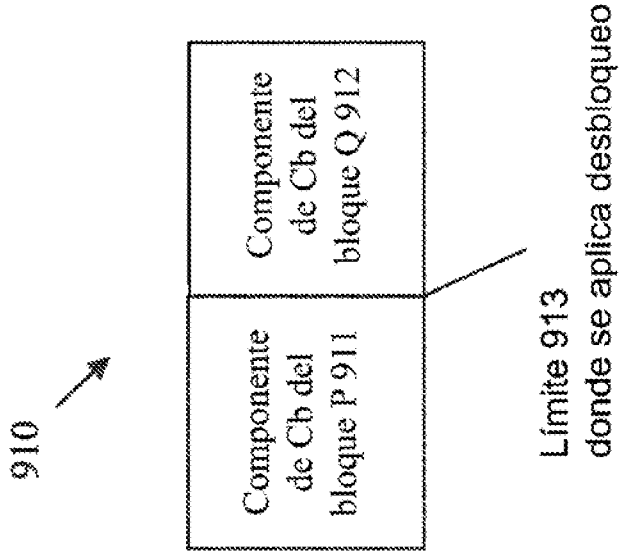
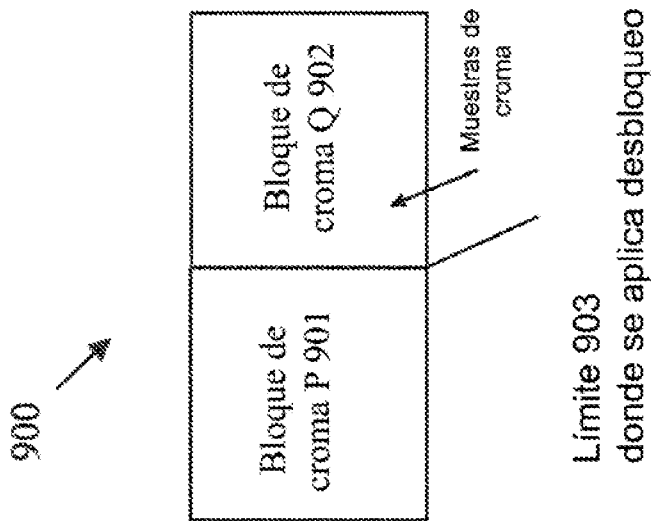


Figura 8



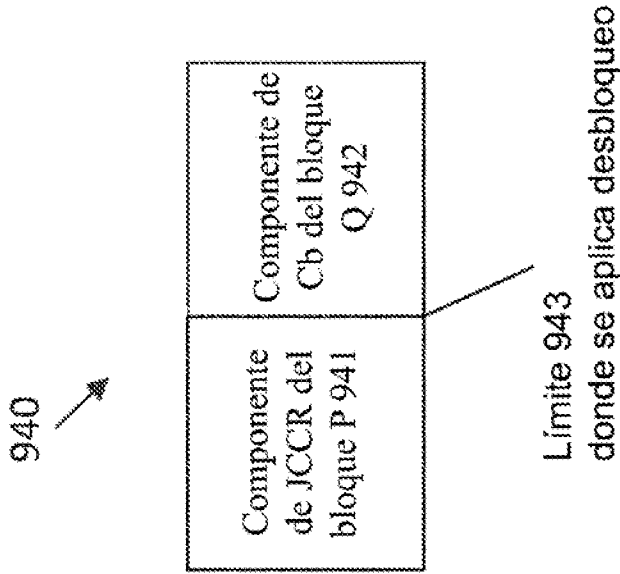


Figura 9E

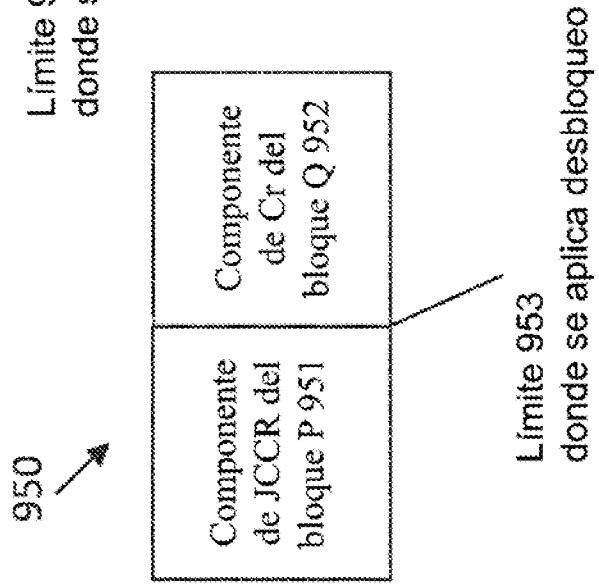


Figura 9F

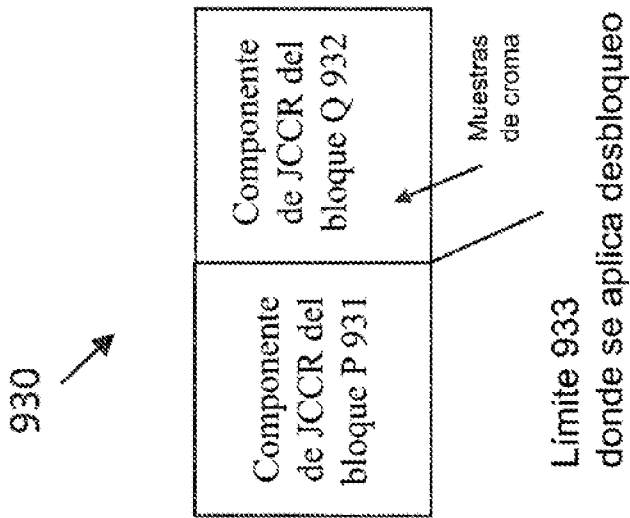


Figura 9D

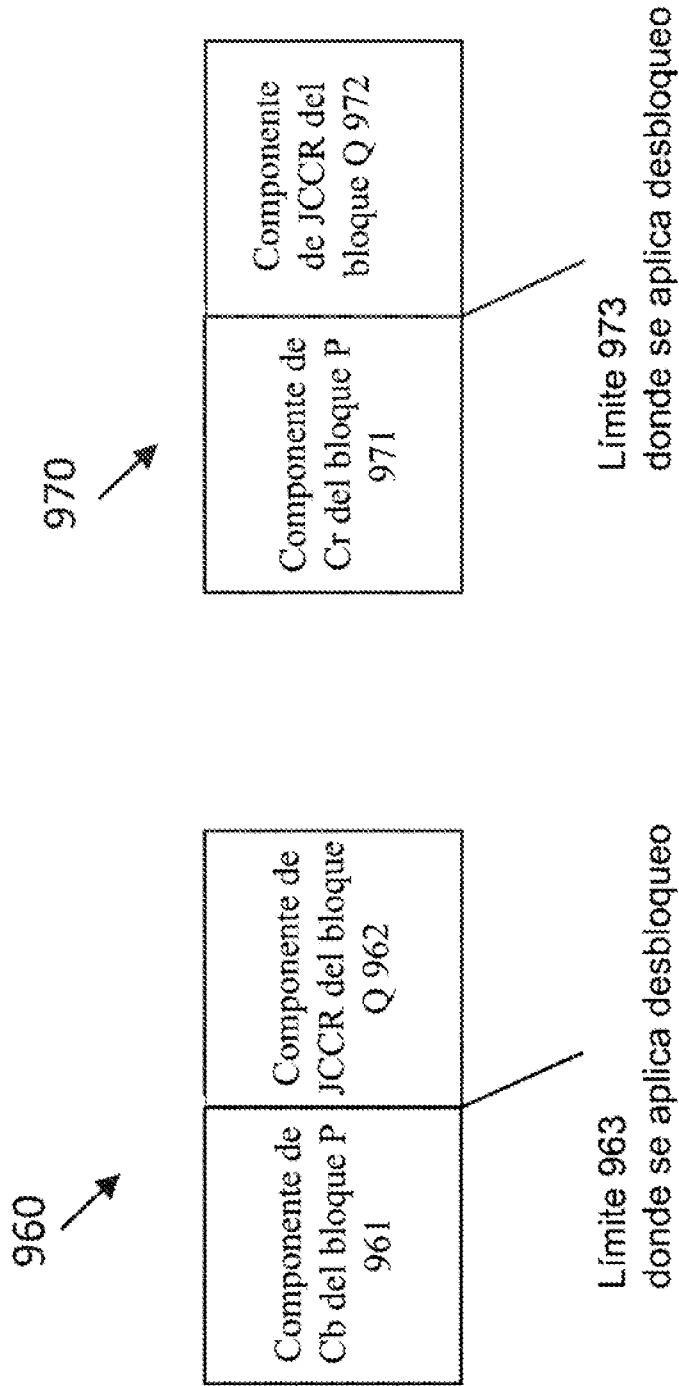


Figura 9G

Figura 9H

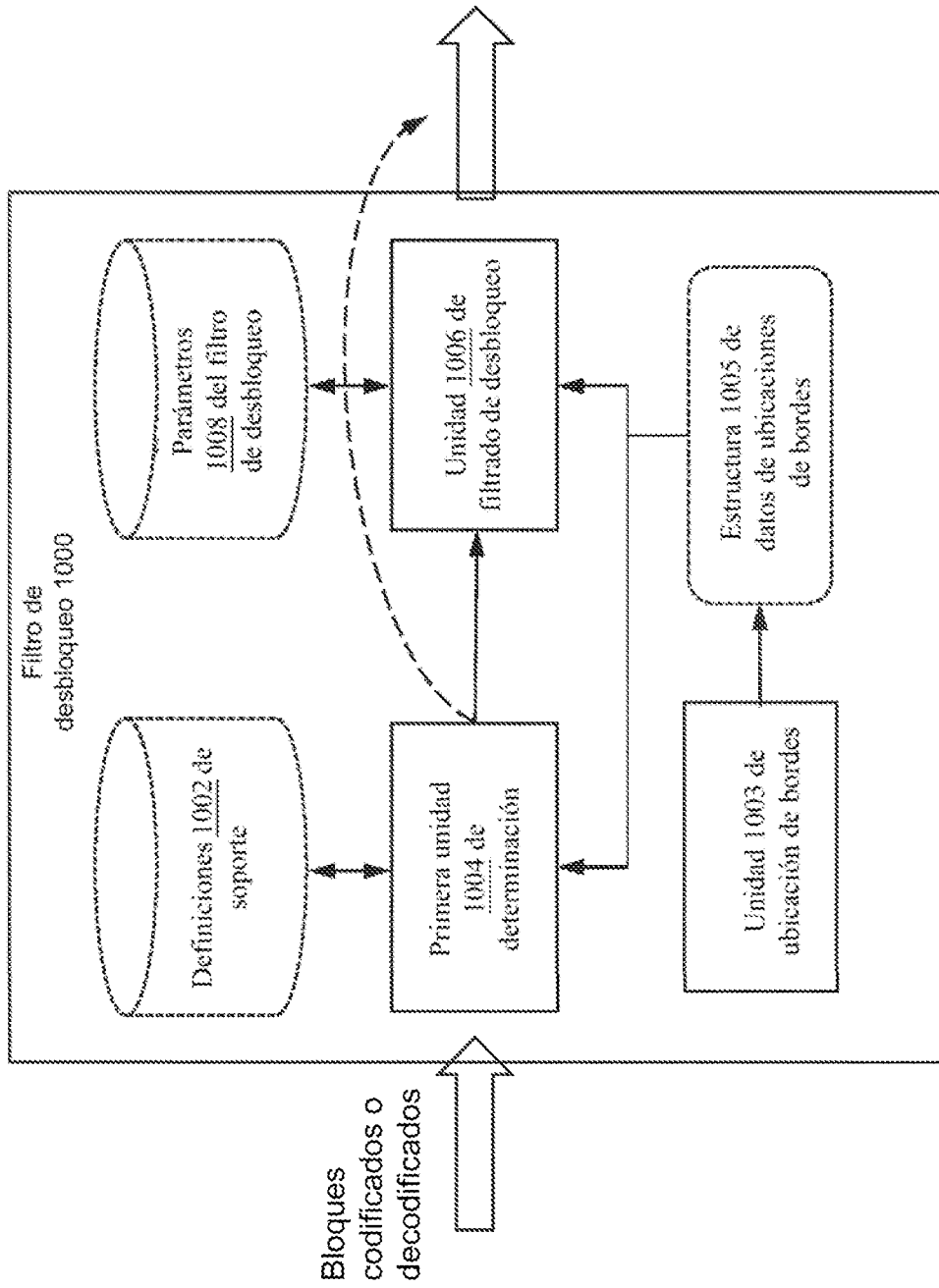


Figura 10

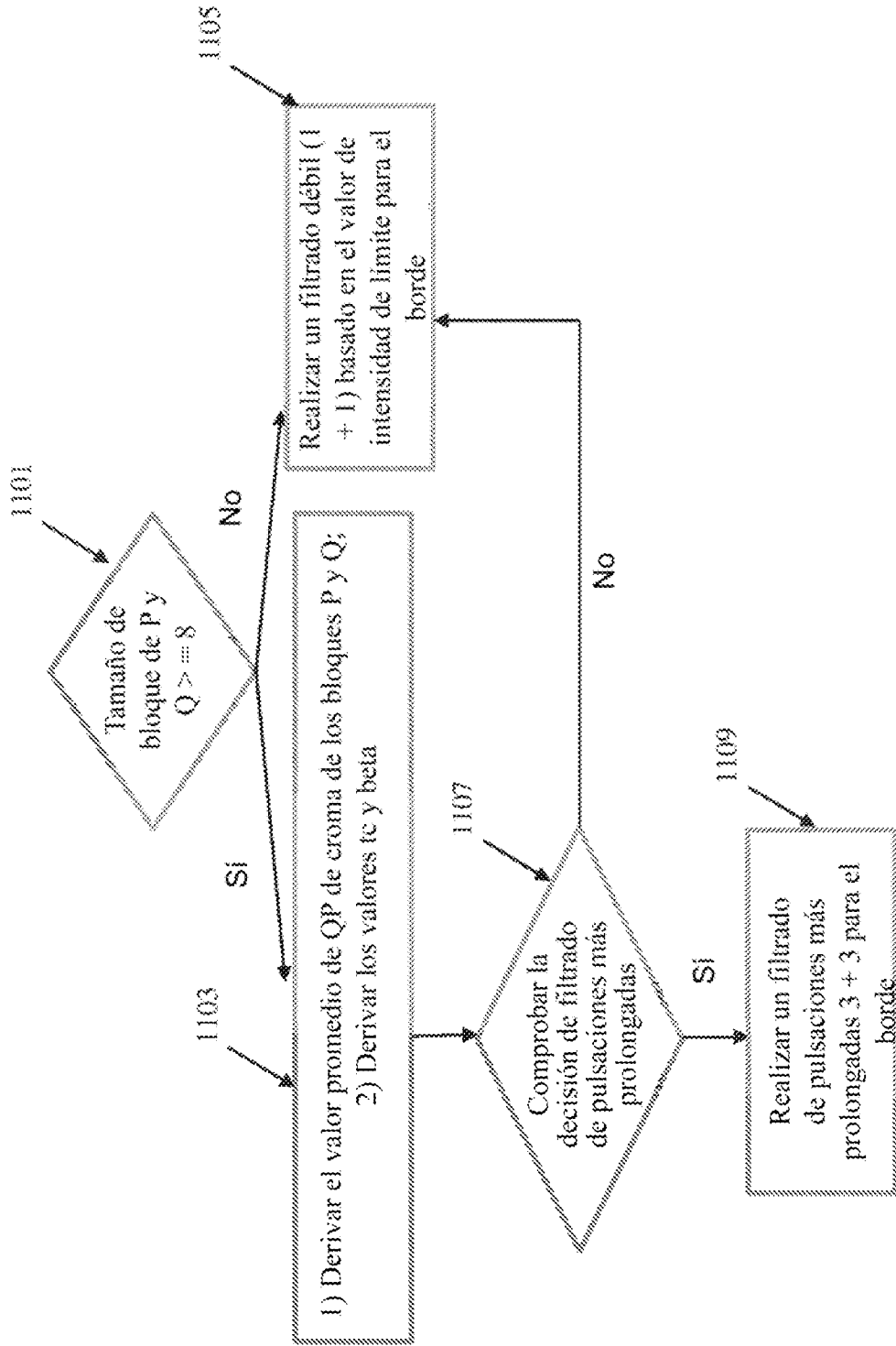


Figura 11

qPi (QP de luma) < 30	30	31	32	33	34	35	36	37	38	39	40	41	42	43	> 43
(0..30)															(44..63)
QP _{chrom} (Clas = 1) = qPi	29	30	31	32	33	33	34	34	35	35	36	36	37	37	= qPi - 6
QP _{chrom} (Clas = 2) = qPi	29	30	31	32	33	33	34	34	35	35	36	36	37	37	= qPi - 6
QP _{chrom} (Clas = 3) = qPi	29	30	31	32	33	36	34	34	35	35	34	36	37	37	= qPi - 7

Tabla 1200

ChromaQpTable[i][k]

Figura 12A

qPi (QP de luma)	< 30 (0 ..30)	30	31	32	33	34	35	36	37	38	39	40	41	42	43	> 43 (44 ..63)
Índice = 0	= qPi	29	30	31	32	33	33	34	34	35	35	36	36	37	37	= qPi - 6
Índice = 1	= qPi	29	30	31	32	33	33	34	34	35	35	36	36	37	37	= qPi - 6
Índice = 2	= qPi	29	30	31	32	33	36	34	34	35	35	34	36	37	37	= qPi - 7

Tabla 1200'

ChromaQpTable[i][k]

Figura 12B

qPi (QP de luma)	< 30	30	31	32	33	34	35	36	37	38	39	40	41	42	43	> 43
	(0 ..30)															(44 ..63)
QP _{cb} (Cb _{cb} == 1)	= qPi	29	30	31	32	33	33	34	34	35	35	36	36	37	37	= qPi - 6

Tabla 1301 de mapeo de croma (Cb)

qPi (QP de luma)	< 30	30	31	32	33	34	35	36	37	38	39	40	41	42	43	> 43
	(0 ..30)															(44 ..63)
QP _{cr} (Cb _{cr} == 2)	= qPi	27	30	31	32	33	33	34	34	35	35	36	36	37	37	= qPi - 7

Tabla 1302 de mapeo de croma (Cr)

qPi (Qpj de luma)	< 30	30	31	32	33	34	35	36	37	38	39	40	41	42	43	> 43
	(0 ..30)															(44 ..63)
QP _{ccr} (Cb _{ccr} == 3)	= qPi	28	30	31	32	33	33	36	36	34	35	36	36	37	37	= qPi - 8

Tabla 1303 de mapeo de Cb-Cr conjunto

Figura 13

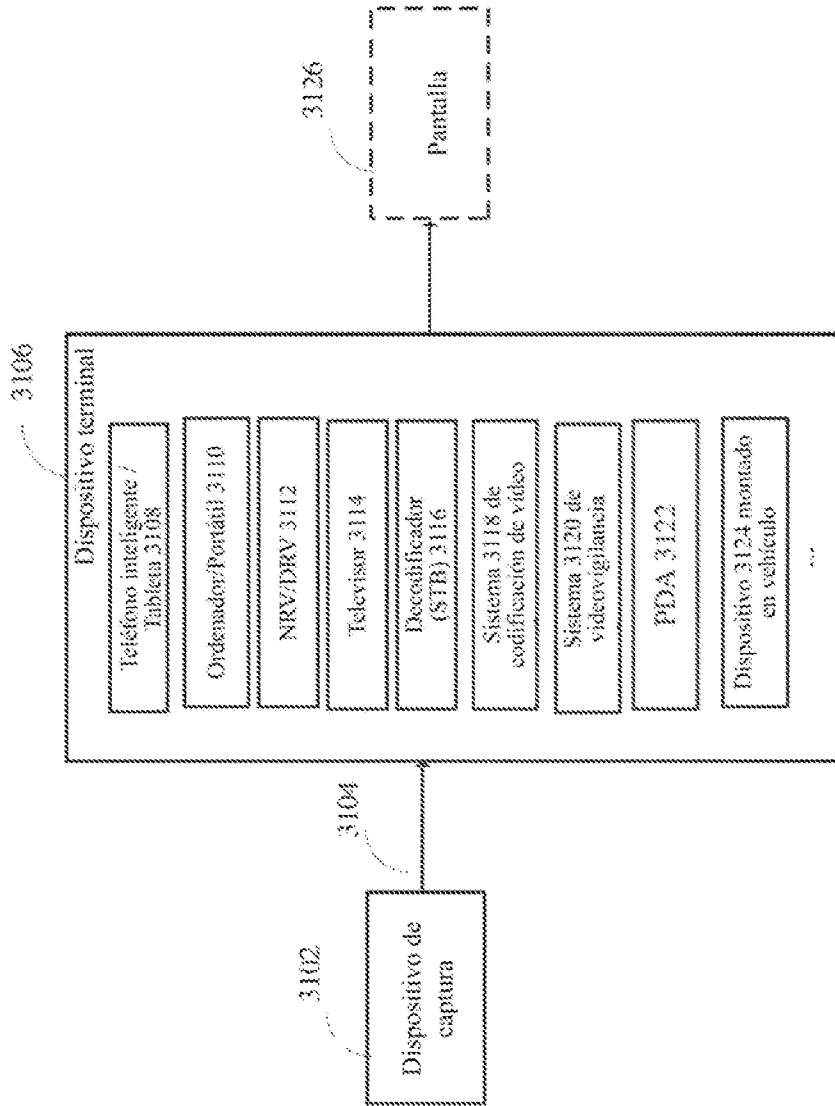


Figura 14

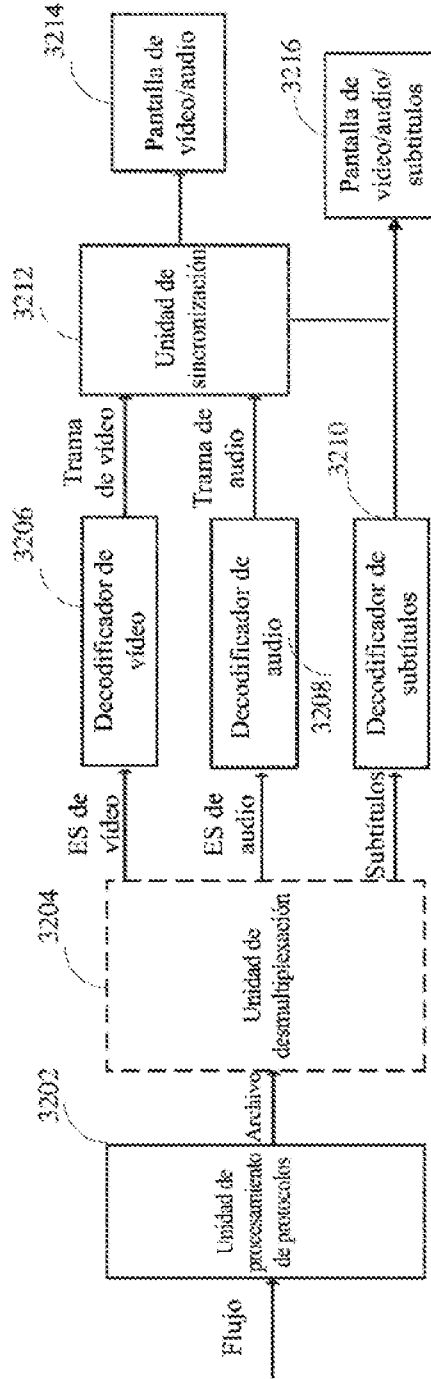


Figura 15

1600

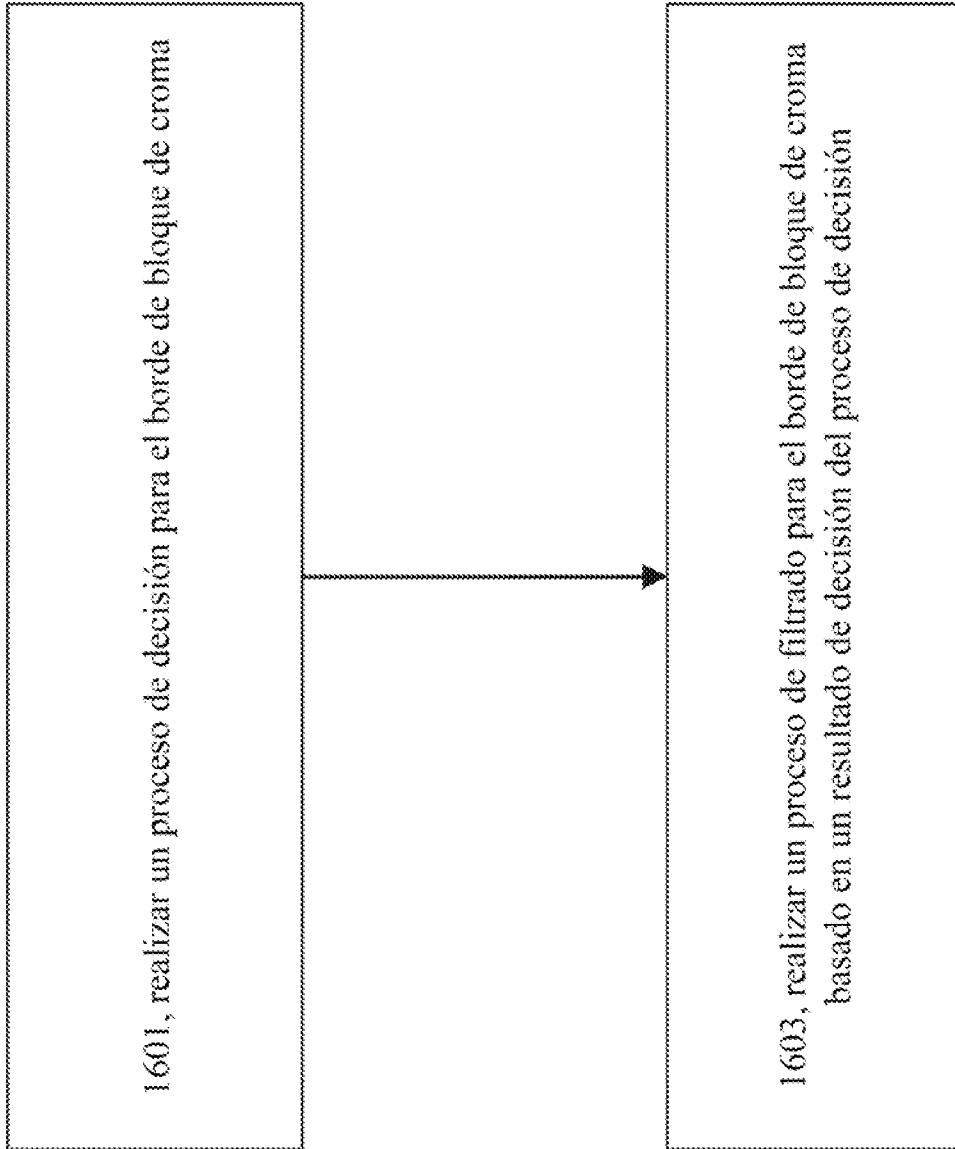


Figura 16

1601

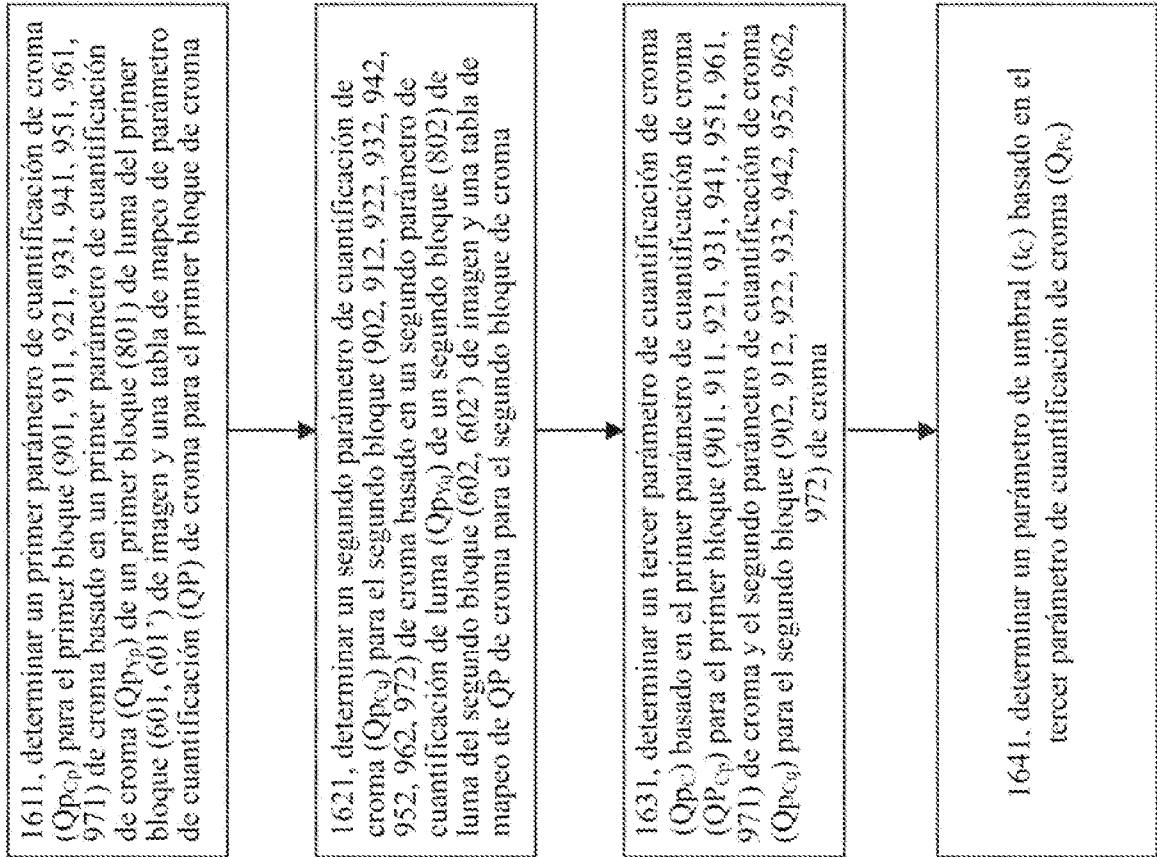


Figura 17