

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 993 252**

51 Int. Cl.:

H04N 19/52 (2014.01)

H04N 19/593 (2014.01)

H04N 19/70 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **10.02.2020** **PCT/CN2020/074575**

87 Fecha y número de publicación internacional: **10.09.2020** **WO20177505**

96 Fecha de presentación y número de la solicitud europea: **10.02.2020** **E 20766336 (0)**

97 Fecha y número de publicación de la concesión europea: **31.07.2024** **EP 3878181**

54 Título: **Un codificador, un decodificador, y métodos correspondientes utilizando lista de fusión IBC**

30 Prioridad:

04.03.2019 US 201962813690 P
07.03.2019 US 201962815311 P

45 Fecha de publicación y mención en BOPI de la
traducción de la patente:
26.12.2024

73 Titular/es:

HUAWEI TECHNOLOGIES CO., LTD. (100.00%)
Huawei Administration Building, Bantian
Longgang District, Shenzhen, Guangdong
518129, CN

72 Inventor/es:

GAO, HAN;
ESENLIK, SEMIH;
WANG, BIAO;
KOTRA, ANAND MEHER y
CHEN, JIANLE

74 Agente/Representante:

PONS ARIÑO, Ángel

ES 2 993 252 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 utilizando lista de fusión IBC

5 Campo técnico

Realizaciones de la presente solicitud (descripción) generalmente se refieren al campo del procesamiento de imágenes y de manera más particular a la lista compartida para predicción.

10 Antecedentes

La codificación de video (codificación y decodificación de video) es utilizada en un amplio intervalo de aplicaciones de video digital, por ejemplo, TV digital de transmisión, transmisión de video sobre Internet y redes móviles, aplicaciones de conversación en tiempo real tales como chat de video, conferencias de video, discos DVD y Blu-ray, sistemas de adquisición y edición de contenido de video, grabadoras de video de aplicaciones de seguridad.

La cantidad de datos de video necesarios para mostrar incluso un video relativamente corto puede ser sustancial, lo cual puede tener como resultado dificultades cuando los datos han de ser transmitidos por secuencias o comunicados de otra manera a través de una red de comunicaciones con capacidad limitada de ancho de banda. Por lo tanto, los datos de video son comprimidos generalmente antes de ser comunicados a través de modem, redes de telecomunicaciones de hoy en día. El tamaño de un video también podría ser un problema cuando el video es almacenado en un dispositivo de almacenamiento debido a que los recursos de memoria pueden ser limitados. Dispositivos de compresión de video con frecuencia utilizan software y/o hardware en la fuente para codificar los datos de video antes de la transmisión o almacenamiento, disminuyendo de esta manera la cantidad de datos necesarios para representar imágenes de video digitales. Los datos comprimidos son recibidos luego en el destino por un dispositivo de descompresión de video que decodifica los datos de video. Con recursos de red limitados y demandas cada vez mayores de calidad de video superior, son deseables técnicas de compresión y descompresión perfeccionadas que mejoren la relación de compresión con poco a ningún sacrificio en la calidad de imagen.

ZHANG (BYTEDANCE) L ET AL: "CE4-related: Restrictions on History-based Motion Vector Prediction", no. JVET-M0272, 17 de enero de 2019 (2019-01-17), XP030202419 describe un método de restricciones para la Predicción de Vectores de Movimiento Basada en el Histórico (HMVP).

WEI-JUNG CHIEN ET AL: "CE4-related: Modification on History-based Mode Vector Prediction", no. JVET-L0401; JVET-L0401, 5 de octubre de 2018 (2018-10-05), páginas 1 a 5, XP030194900 describe un método para reducir la complejidad del modo HMVP.

HAN (QUALCOMM) Y ET AL: "CE4: Modification on History-based Motion Vector Prediction", no. JVET-M0126, 19 de enero de 2019 (2019-01-19), XP030202546, describe un método para actualizar la tabla HMVP.

El documento EP3308544A1 describe un método de codificación de datos de video que incluye la determinación de uno o más parámetros de compensación de iluminación para un bloque actual y la codificación del bloque actual como parte de una corriente de bits codificada utilizando el uno o más parámetros de compensación de iluminación

Compendio

Realizaciones de la presente solicitud proporcionan aparatos y métodos para codificación y decodificación según las reivindicaciones independientes.

Los objetivos anteriores así como otros se logran mediante el tema de las reivindicaciones independientes. Formas de implementación adicionales son evidentes a partir de las reivindicaciones dependientes, de la descripción y las figuras. En la siguiente descripción, las realizaciones no cubiertas por las reivindicaciones deben considerarse como ejemplos útiles para comprender la invención

Detalles de una o más realizaciones se exponen en los dibujos adjuntos y en la siguiente descripción. Otras características, objetivos y ventajas serán evidentes a partir de la descripción, dibujos y reivindicaciones.

60 Breve descripción de los dibujos

A continuación, se describen con más detalle realizaciones de la invención con referencia a las figuras y dibujos adjuntos, en los cuales:

La figura 1A es un diagrama de bloques que muestra un ejemplo de un sistema de codificación de video

configurado para implementar realizaciones de la invención;

La figura 1B es un diagrama de bloques que muestra otro ejemplo de un sistema de codificación de video configurado para implementar realizaciones de la invención;

5

La figura 2 es un diagrama de bloques que muestra un ejemplo de un codificador de video configurado para implementar realizaciones de la invención;

10

La figura 3 es un diagrama de bloques que muestra una estructura ejemplar de un decodificador de video configurado para implementar realizaciones de la invención;

La figura 4 es un diagrama de bloques que ilustra un ejemplo de un aparato de codificación o un aparato de decodificación;

15

La figura 5 es un diagrama de bloques que ilustra otro ejemplo de un aparato de codificación o un aparato de decodificación;

La figura 6 es un diagrama de bloques que ilustra un ejemplo de bloques contiguos de un bloque actual.

20

La figura 7 muestra un ejemplo de la presente solicitud.

La figura 8 es un diagrama de bloques que muestra una estructura ejemplar de un sistema 3100 de suministro de contenido que lleva a cabo un servicio de entrega de contenido.

25

La figura 9 es un diagrama de bloques que muestra una estructura de un ejemplo de un dispositivo terminal.

En lo sucesivo, signos de referencia idénticos hacen referencia a características idénticas o al menos funcionalmente equivalentes si no se especifica explícitamente de otra manera.

30

Descripción detallada de las realizaciones

En la siguiente descripción, se hace referencia a las figuras adjuntas, que forman parte de la descripción, y las cuales muestran, a manera de ilustración, aspectos específicos de realizaciones de la invención o aspectos específicos en los cuales se pueden utilizar las realizaciones de la presente invención. Se entiende que realizaciones de la invención pueden ser utilizadas en otros aspectos y comprenden cambios estructurales o lógicos no mostrados en las figuras. La siguiente descripción detallada, por tanto, no se ha de tomar en un sentido limitativo, y el alcance de la presente invención es definido por las reivindicaciones adjuntas.

35

Por ejemplo, se entiende que una descripción en conexión con un método descrito también puede ser verdadera para un dispositivo o sistema correspondiente configurado para realizar el método y viceversa. Por ejemplo, si se describe una o una pluralidad de etapas de método específicas, un dispositivo correspondiente puede incluir una o una pluralidad de unidades, por ejemplo unidades funcionales, para realizar la etapa o pluralidad de etapas de método descritas (por ejemplo una unidad que realiza una o una pluralidad de etapas, o una pluralidad de unidades, cada una las cuales realiza una o más de la pluralidad de etapas), incluso si dicha unida o unidades no son explícitamente descritas o ilustradas en las figuras. Por otra parte, por ejemplo, si un aparato específico es descrito basándose 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 una pluralidad de unidades (por ejemplo una etapa que realiza la funcionalidad de una o una pluralidad de unidades, o una pluralidad de etapas, realizando cada una la funcionalidad de una o más de la pluralidad de unidades), incluso si dicha etapa o pluralidad de etapas no se describen o ilustran explícitamente en las figuras. Además, se entiende que las características de las distintas realizaciones y/o aspectos ejemplares descritos en la presente memoria se pueden combinar entre sí, a menos que específicamente se indique de otra manera.

40

45

50

55

60

65

La codificación de video típicamente se refiere al procesamiento de una secuencia de imágenes, que forman el video o la secuencia de video. En lugar del término "imagen" se puede utilizar el término "fotograma" o "fotografía" como sinónimos en el campo de la codificación de video. La codificación de video (o codificación en general) comprende dos partes, codificación de video y decodificación de video. La codificación de video es realizada en el lado fuente, comprendiendo típicamente el procesamiento (por ejemplo por compresión) de las imágenes de video originales para reducir la cantidad de datos requeridos para representar las imágenes de video (para almacenamiento y/o transmisión más eficientes). La decodificación de video es realizada en el lado destino y típicamente comprende el procesamiento inverso en comparación con el codificador para reconstruir las imágenes de video. Realizaciones que hacen referencia a "codificación" de imágenes de video (o imágenes en general) se entenderán como relacionadas con "codificación" o "decodificación" de imágenes de video o secuencias de video respectivas. La combinación de la parte de codificación y la parte de decodificación es también denominada como CODEC (Codificación y Decodificación).

En caso de codificación de video sin pérdida, las imágenes de video originales pueden ser reconstruidas, es decir las imágenes de video reconstruidas tienen la misma calidad que las imágenes de video originales (asumiendo que no hay pérdida de transmisión u otra pérdida de datos durante el almacenamiento o transmisión). En caso de decodificación de video con pérdida, se realiza compresión adicional, por ejemplo mediante cuantificación, para reducir la cantidad de datos que representan las imágenes de video, las cuales no pueden ser completamente reconstruidas en el decodificador, es decir la calidad de las imágenes de video reconstruidas es inferior o peor en comparación con la calidad de las imágenes de video originales.

Varias normas de codificación de video pertenecen al grupo de "códecs de video híbrido con pérdida" (es decir combinan predicción espacial y temporal en el dominio de muestra y codificación de transformada 2D para aplicar cuantificación en el dominio de transformada). Cada imagen de una secuencia de video típicamente es dividida en particiones en un conjunto de bloques que no se superponen y la codificación típicamente es realizada en un nivel de bloque. En otras palabras, en el codificador el video típicamente es procesado, es decir codificado, en un nivel de bloque (bloque de video), por ejemplo utilizando predicción espacial (intra-imagen) y/o predicción temporal (inter-imagen) para generar un bloque de predicción, sustrayendo el bloque de predicción del bloque actual (bloque actualmente procesado/que ha de ser procesado) 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 que se han de transmitir (compresión), mientras que en el decodificador se aplica el procesamiento inverso, en comparación con el codificador, al bloque codificado o comprimido para reconstruir el bloque actual para representación. Además, el codificador duplica el bucle de procesamiento del decodificador de manera que ambos generarán predicciones idénticas (por ejemplo intra-predicciones e inter-predicciones) y/o reconstrucciones para procesar, es decir codificar, los bloques subsiguientes.

A continuación se describen realizaciones de un sistema 10 de codificación de video, un codificador 20 de video y un decodificador 30 de video basándose en las figuras 1 a 3.

La figura 1A es un diagrama de bloques esquemático que ilustra un sistema 10 de codificación ejemplar, por ejemplo un sistema 10 de codificación de video (o para abreviar sistema 10 de codificación) que puede utilizar técnicas de esta presente solicitud. El codificador 20 de video (o para abreviar codificador 20) y el decodificador 30 de video (o para abreviar decodificador 30) del sistema 10 de codificación de video representan ejemplos de dispositivos que pueden ser configurados para realizar técnicas de acuerdo con diversos ejemplos descritos en la presente solicitud.

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

El dispositivo fuente 12 comprende un codificador 20, y adicionalmente, es decir opcionalmente, puede comprender una fuente 16 de imagen, un pre-procesador 18 (o unidad de pre-procesamiento), por ejemplo un pre-procesador 18 de imagen, y una interfaz de comunicación o unidad 22 de comunicación.

La fuente 16 de imagen puede comprender o puede ser cualquier tipo de dispositivo de captura de imagen, por ejemplo una cámara para capturar una imagen del mundo real y/o cualquier tipo de dispositivo de generación de imagen, por ejemplo un procesador de gráficos por 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 (por ejemplo un contenido de pantalla, una imagen de realidad virtual (RV)) y/o cualquier combinación de los mismos (por ejemplo una imagen de realidad aumentada (RA)). La fuente de imagen puede ser cualquier tipo de memoria o almacenamiento que almacene cualquiera de las imágenes antes mencionadas.

A diferencia del pre-procesador 18 y del procesamiento realizado por la unidad 18 de pre-procesamiento, la imagen o datos 17 de imagen también pueden ser denominados como imagen sin procesar o datos 17 de imagen sin procesar.

El pre-procesador 18 está configurado para recibir los datos 17 e imagen (sin procesar) y para realizar pre-procesamiento en los datos 17 de imagen para obtener una imagen 19 pre-procesada o datos 19 de imagen pre-procesada. El pre-procesamiento ejecutado por el pre-procesador 18, por ejemplo, puede comprender recorte, conversión de formato de color (por ejemplo de RGB a YCbCr), corrección de color, o eliminación de ruido. Se puede entender que la unidad 18 de pre-procesamiento puede ser un componente opcional.

El codificador 20 de video está configurado para recibir los datos 19 de imagen pre-procesada y puede proporcionar datos 21 de imagen codificados(a continuación se describirán detalles adicionales, por ejemplo, basándose en la figura 2).

La interfaz 22 de comunicación del dispositivo fuente 12 se puede configurar para recibir los datos 21 de imagen codificados y transmitir los datos 21 de imagen codificados (o cualquier versión procesada adicional de los

misimos) sobre el canal 13 de comunicación a otro dispositivo, por ejemplo el dispositivo destino 14 o cualquier otro dispositivo, para almacenamiento o reconstrucción directa.

5 El dispositivo destino 14 comprende un decodificador 30 (por ejemplo un decodificador 30 de video), y adicionalmente, es decir opcionalmente, puede comprender una interfaz de comunicación o unidad 28 de comunicación, un post-procesador 32 (o una unidad 32 de post-procesamiento) y un dispositivo 34 de visualización.

10 La interfaz 28 de comunicación del dispositivo destino 14 está configurada para recibir los datos 21 de imagen codificados (o cualquier versión procesada adicional de los mismos), por ejemplo directamente desde el dispositivo fuente 12 o desde cualquier otra fuente, por ejemplo un dispositivo de almacenamiento, por ejemplo un dispositivo de almacenamiento de datos de imagen codificados, y proporcionar los datos 21 de imagen codificados al decodificador 30.

15 La interfaz 22 de comunicación y la interfaz 28 de comunicación pueden ser configuradas para transmitir o recibir los datos 21 de imagen codificados o los datos codificados 13 a través de un enlace de comunicación directa entre el dispositivo fuente 12 y el dispositivo destino 14, por ejemplo una conexión cableada o inalámbrica directa, 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.

20 La interfaz 22 de comunicación por ejemplo, puede ser configurada para empaquetar los datos 21 de imagen codificados en un formato apropiado, por ejemplo paquetes, y/o procesar los datos de imagen codificados utilizando cualquier tipo de codificación de transmisión o procesamiento para transmisión sobre un enlace de comunicación o red de comunicación.

25 La interfaz 28 de comunicación, que forma la contraparte de la interfaz 22 de comunicación, por ejemplo, puede ser configurada para recibir los datos transmitidos y procesar los datos de transmisión utilizando cualquier tipo de decodificación o procesamiento de transmisión correspondiente y/o desempaquetando para obtener datos 30 21 de imagen codificados.

35 Ambas, la interfaz 22 de comunicación y la interfaz 28 de comunicación pueden ser configuradas como interfaces de comunicación unidireccionales conforme a lo indicado por la flecha para el canal 13 de comunicación en la figura 1A apuntando desde el dispositivo fuente 12 al dispositivo destino 14, o interfaces de comunicación bidireccionales, y se pueden configurar, por ejemplo para enviar y recibir mensajes, por ejemplo para establecer una conexión, para reconocer e intercambiar cualquier otra información relacionada con el enlace de comunicación y/o transmisión de datos, por ejemplo transmisión de datos de imagen codificados.

40 El decodificador 30 está configurado para recibir los datos 21 de imagen codificados y proporcionar los datos 31 de imagen decodificados o una imagen decodificada 31 (detalles adicionales se describirán a continuación, por ejemplo, basándose en la figura 3 o en la figura 5).

45 El post-procesador 32 del dispositivo destino 14 está configurado para post-procesar los datos 31 de imagen decodificados (también denominados datos de imagen reconstruidos), por ejemplo la imagen decodificada 31, para obtener datos 33 de imagen post-procesada, por ejemplo una imagen post-procesada 33. El post-procesamiento ejecutado por la unidad 32 de post-procesamiento puede comprender, por ejemplo conversión de formato de color (por ejemplo de YCbCr a RGB), corrección de color, recorte, o nuevo muestreo, o cualquier otro procesamiento, por ejemplo para preparar los datos 31 de imagen decodificados para visualización, por ejemplo a través del dispositivo 34 de visualización.

50 El dispositivo 34 de visualización del dispositivo destino 14 está configurado para recibir los datos 33 de imagen post-procesada para mostrar la imagen, por ejemplo a un usuario o un espectador. El dispositivo 34 de visualización puede ser o comprender cualquier tipo de dispositivo de visualización para representar la imagen reconstruida, por ejemplo un dispositivo de visualización o monitor integrado o externo. Los dispositivos de visualización, por ejemplo pueden comprender pantallas de cristal líquido (LCD), pantallas de diodos orgánicos de emisión de luz (OLED), pantallas de plasma, proyectores, micro-pantallas de LED, cristal líquido en silicio (LCoS), procesador de luz digital (DLP) o cualquier tipo de otra pantalla.

55 Aunque la figura 1A muestra el dispositivo fuente 12 y el dispositivo destino 14 como dispositivos separados, realizaciones de los dispositivos también pueden comprender ambos o ambas funcionalidades, el dispositivo fuente 12 o la funcionalidad correspondiente y el dispositivo destino 14 o la funcionalidad correspondiente. En dichas realizaciones el dispositivo fuente 12 o la funcionalidad correspondiente y el dispositivo destino 14 o la funcionalidad correspondiente pueden ser implementados utilizando el mismo hardware y/o software o mediante hardware y/o software separados o cualquier combinación de los mismos.

Como resultará evidente para un experto en la técnica basándose en la descripción, la existencia y división (exacta) de funcionalidades de las diferentes unidades o funcionalidades dentro del dispositivo fuente 12 y/o dispositivo destino 14 como se muestra en la figura 1A pueden variar dependiendo del dispositivo y aplicación reales.

5

El codificador 20 (por ejemplo un codificador 20 de video) o el decodificador 30 (por ejemplo un decodificador 30 de video) o tanto el codificador 20 como el decodificador 30 pueden ser implementados a través de circuitos de procesamiento como se muestra en la figura 1B, tal como uno o más microprocesadores, procesadores de señal digital (DSP), circuitos integrados de aplicación específica (ASIC), matrices de puertas programables en campo (FPGA), lógica discreta, hardware, codificación de video dedicada o cualquier combinación de los mismos. El codificador 20 puede ser implementado a través de circuitos 46 de procesamiento para realizar los diversos módulos como se analiza 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 ser implementado a través de circuitos 46 de procesamiento para incorporar los diversos módulos como se analizó con respecto al decodificador 30 de la figura 3 y/o cualquier otro sistema o subsistema decodificador descrito en la presente memoria. Los circuitos de procesamiento pueden ser configurados para ejecutar las diversas operaciones como se analiza más adelante. Como se muestra en la figura 5, si las técnicas son implementadas parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio de almacenamiento legible por ordenador no-transitorio, adecuado y puede ejecutar las instrucciones en hardware utilizando uno o más procesadores para realizar las técnicas de esta descripción. Cualquiera del codificador 20 de video y del decodificador 30 de video puede ser integrado como parte de un codificador/decodificador combinado (CODEC) en un solo dispositivo, por ejemplo, como se muestra en la figura 1B.

10

15

20

25

30

El dispositivo fuente 12 y el dispositivo destino 14 pueden comprender cualquiera de una amplia variedad de dispositivos, incluyendo cualquier tipo de dispositivos manuales o estacionarios, por ejemplo ordenadores portátiles o laptop, teléfonos móviles, teléfonos inteligentes, tabletas u ordenadores de tableta, cámaras, ordenadores de escritorio, decodificadores, televisiones, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de transmisión por secuencias de video (tal como servidores de servicios de contenido o servidores de entrega de contenido), dispositivo receptor de difusión, dispositivo transmisor de difusión, o similar y puede no utilizar o utilizar cualquier tipo de sistema operativo. En algunos casos, el dispositivo fuente 12 y el dispositivo destino 14 pueden estar equipados para comunicación inalámbrica. Por lo tanto, el dispositivo fuente 12 y el dispositivo destino 14 pueden ser dispositivos de comunicación inalámbrica.

35

40

En algunos casos, el sistema de codificación de video 10 ilustrado en la figura 1A es simplemente un ejemplo y las técnicas de la presente solicitud se pueden aplicar a ajustes de codificación de video (por ejemplo, codificación de video o decodificación de video) que no necesariamente incluyen alguna comunicación de datos entre los dispositivos de codificación y decodificación. En otros ejemplos, los datos son recuperados de una memoria local, transmitidos sobre una red, o similar. Un dispositivo de codificación de video puede codificar y almacenar datos en la memoria, y/o un dispositivo de decodificación de video puede recuperar y decodificar datos desde la memoria. En algunos ejemplos, la codificación y decodificación es ejecutada por dispositivos que no se comunican entre sí, sino que simplemente codifican datos en la memoria y/o recuperan y decodifican datos desde la memoria.

45

50

Por conveniencia de la descripción, en la presente memoria se describen realizaciones de la invención, por ejemplo, por referencia a Codificación de Video de Alta Eficiencia (HEVC) o al software de referencia de la Codificación de Video Versátil (VVC), la norma de codificación de video de siguiente generación desarrollada por el Equipo de Colaboración Conjunto en Codificación de Video (JCT-VC) del Grupo de Expertos de Codificación de Video ITU-T (VCEG) y del Grupo de Expertos de Imágenes en Movimiento ISO/IEC (MPEG). Un experto en la técnica entenderá que realizaciones de la invención no se limitan a HEVC o VVC.

Codificador y método de codificación

55

60

65

La figura 2 muestra un diagrama de bloques esquemático de un codificador 20 de video ejemplar que está configurado para implementar las técnicas de la presente solicitud. En el ejemplo de la figura 2, el codificador 20 de video 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, y una unidad 212 de procesamiento de transformada inversa, una unidad 214 de reconstrucción, una unidad 220 de filtro de bucle, una memoria tampón 230 de imágenes decodificadas (DPB), 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 inter-predicción, una unidad 254 de intra-predicción y una unidad 262 de partición. La unidad 244 de inter-predicción puede incluir una unidad de estimación de movimiento y una unidad de compensación de movimiento (que no se muestra). Un codificador 20 de video tal como se muestra en la figura 2 también se puede denominar como codificador de video híbrido o un codificador de video de acuerdo con un códec de video híbrido.

La unidad 204 de cálculo residual, la unidad 206 de procesamiento de transformada, la unidad 208 de cuantificación, la unidad 260 de selección de modo, pueden ser denominadas como formando una trayectoria de señal de avance del codificador 20, mientras que la unidad 210 de cuantificación inversa, la unidad 212 de procesamiento de transformada inversa, la unidad 214 de reconstrucción, la memoria tampón 216, el filtro 220 de bucle, la memoria tampón 230 de imágenes decodificadas (DPB), la unidad 244 de inter-predicción y la unidad 254 de intra-predicción pueden ser denominadas como formando una trayectoria de señal de retorno del codificador 20 de video, en donde la trayectoria de señal de retorno del codificador 20 de video corresponde a la trayectoria de señal del decodificador (véase decodificador 30 de video en la figura 3). 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 tampón 230 de imágenes decodificadas (DPB), la unidad 244 de inter-predicción y la unidad 254 de intra-predicción también son denominadas como formando el "decodificador integrado" del codificador 20 de video.

Imágenes y División en Particiones de Imágenes (Imágenes y Bloques)

El codificador 20 puede estar configurado para recibir, por ejemplo a través de la entrada 201, una imagen 17 (o datos 17 de imagen), por ejemplo imagen de una secuencia de imágenes que forman un video o secuencia de video. La imagen o datos de imagen recibidos también pueden ser una imagen pre-procesada 19 (o datos 19 de imagen pre-procesada). Con propósito de simplicidad la siguiente descripción se refiere a la imagen 17. La imagen 17 también se puede denominarse como imagen actual o imagen que ha de ser codificada (en particular en la codificación de video para distinguir la imagen actual de otras imágenes, por ejemplo imágenes previamente codificadas y/o decodificadas de la misma secuencia de video, es decir la secuencia de video que también comprende la imagen actual).

Una imagen (digital) es o puede ser vista como una agrupación o matriz bidimensional de muestras con valores de intensidad. Una muestra en la agrupación también se puede denominar como pixel (forma abreviada de elemento de imagen) o un pel. El número de muestras en la dirección (o eje) horizontal y vertical de la agrupación o imagen definen el tamaño y/o resolución de la imagen. Para representación de color, típicamente se emplean tres componentes de color, es decir la imagen puede ser representada o puede incluir tres agrupaciones de muestras. En formato o espacio de color RGB una imagen comprende una agrupación de muestra roja, verde y azul correspondiente. No obstante, en la codificación de video, cada pixel es representado típicamente en un formato o espacio de color de luminancia y crominancia, por ejemplo YCbCr, el cual comprende un componente de luminancia indicado por Y (en ocasiones también se utiliza L en su lugar) y dos componentes de crominancia indicados por Cb y Cr. El componente Y de luminancia (o luma para abreviar) representa el brillo o intensidad del nivel de grises (por ejemplo tal como en una imagen en escala de grises), mientras que los dos componentes de crominancia (o croma para abreviar) Cb y Cr representan los componentes de cromaticidad o de información de color. Por consiguiente, una imagen en formato YCbCr comprende una agrupación de muestras de luminancia de valores (Y) de muestras de luminancia, y dos agrupaciones de muestras de crominancia de valores de crominancia (Cb y Cr). Imágenes en formato RGB pueden ser convertidas o transformadas en formato YCbCr y viceversa, el proceso también se conoce como transformación o conversión de color. Si una imagen es monocromática, la imagen puede comprender solamente una agrupación de muestras de luminancia. Por consiguiente, una imagen puede ser, por ejemplo, una agrupación de muestras de luminancia en formato monocromático o una agrupación de muestras de luminancia y dos agrupaciones correspondientes de muestras de crominancia en formato de color 4:2:0, 4:2:2, y 4:4:4.

Realizaciones del codificador 20 de video pueden comprender una unidad de división en particiones de imagen (que no se muestra en la figura 2) configurada para dividir en particiones la imagen 17 en una pluralidad de bloques 203 de imagen (que típicamente no se superponen). Estos bloques también se pueden denominar como bloques raíz, macro bloques (H.264/AVC) o bloques de árbol de codificación (CTB) o unidades de árbol de codificación (CTU) (H.265/HEVC y VVC). La unidad de división en particiones de imagen puede ser configurada para utilizar el mismo tamaño de bloque para todas las imágenes de una secuencia de video y la rejilla correspondiente que define el tamaño de bloque, o para cambiar el tamaño de bloque entre imágenes o subconjuntos o grupos de imágenes, y dividir en particiones cada imagen en los bloques correspondientes.

En realizaciones adicionales, el codificador de video puede ser configurado para recibir directamente un bloque 203 de la imagen 17, por ejemplo uno, varios o todos los bloques que forman la imagen 17. El bloque 203 de imagen también puede ser denominado como un bloque de imagen actual o bloque de imagen que ha de ser codificado.

Al igual que la imagen 17, el bloque 203 de imagen de nuevo, es o puede ser considerado como una agrupación o matriz bidimensional de muestras con valores de intensidad (valores muestra), aunque de dimensión menor que la imagen 17. En otras palabras, el bloque 203 puede comprender, por ejemplo, una agrupación de muestras (por ejemplo una agrupación de luma en caso de una imagen monocromática 17, o una agrupación de luma o de croma en caso de una imagen de color) o tres agrupaciones de muestras (por ejemplo una agrupación de luma y dos agrupaciones de croma en el caso de una imagen 17 de color) o cualquier otro

número y/o tipo de agrupaciones dependiendo del formato de color aplicado. El número de muestras en dirección (o eje) horizontal y vertical del bloque 203 define el tamaño de bloque 203. Por consiguiente, un bloque, por ejemplo, puede ser una agrupación MxN de muestras (M-columnas por N-filas), o una agrupación MxN de coeficientes de transformada.

5

Realizaciones del codificador 20 de video, como se muestra en la figura 2, pueden ser configuradas para codificar la imagen 17 bloque por bloque, por ejemplo la codificación y predicción es realizada por bloque 203.

10

Realizaciones del codificador 20 de video, como se muestra en la figura 2, pueden ser configuradas adicionalmente para dividir en particiones y/o codificar la imagen utilizando segmentos (también denominados como segmentos de video), en donde una imagen puede ser dividida en particiones o codificada utilizando uno o más segmentos (típicamente sin superponerse), y cada segmento puede comprender uno o más bloques (por ejemplo CTU).

15

Realizaciones del codificador 20 de video como se muestran en la figura 2 además se pueden configurar para dividir en particiones y/o codificar la imagen utilizando grupos de mosaicos (también denominados como grupos de mosaicos de video) y/o mosaicos (también denominados como mosaicos de video), en donde una imagen puede ser dividida en particiones o codificada utilizando uno o más grupos de mosaicos (típicamente sin superponerse), y cada grupo de mosaicos puede comprender, por ejemplo uno o más bloques (por ejemplo CTU) o uno o más mosaicos, en donde cada mosaico, por ejemplo puede tener forma rectangular y puede comprender uno o más bloques (por ejemplo CTU), por ejemplo bloques completos o fraccionarios.

20

Cálculo residual

25

La unidad 204 de cálculo residual puede ser configurada para calcular un bloque residual 205 (también denominado como residual 205) basándose en el bloque 203 de imagen y un bloque 265 de predicción (más adelante se proporcionan detalles adicionales acerca del bloque 265 de predicción), por ejemplo sustrayendo valores de muestra del bloque 265 de predicción de los valores de muestra del bloque 203 de imagen, muestra por muestra (pixel por pixel) para obtener el bloque residual 205 en el dominio de muestra.

30

Transformada

35

La unidad 206 de procesamiento de transformada puede ser configurada para aplicar una transformada, por ejemplo una transformada de coseno discreta (DCT) o una transformada de seno discreta (DST), sobre los valores muestra del bloque residual 205 para obtener coeficientes 207 de transformada en un dominio de transformada. Los coeficientes 207 de transformada también pueden ser denominados como coeficientes residuales de transformada y representan el bloque residual 205 en el dominio de transformada.

40

La unidad 206 de procesamiento de transformada puede ser configurada para aplicar aproximaciones de números enteros de DCT/DST, tales como las transformadas especificadas para H.265/HEVC. En comparación con una transformada DCT ortogonal, dichas aproximaciones de números enteros típicamente son escaladas por un cierto factor. A fin de preservar la norma del bloque residual el cual es procesado por transformadas directa e inversa, se aplican factores de escalado adicionales como parte del proceso de transformada. Los factores de escalado son elegidos típicamente con base en ciertas restricciones tales como factores de escalado que son una potencia de dos para operaciones de cambio, profundidad de bit de los coeficientes de transformada, compensación entre precisión y costes de implementación, etc. Factores de escalado específicos, por ejemplo, son especificados para la transformada inversa, por ejemplo por la unidad 212 de procesamiento de transformada inversa, (y la transformada inversa correspondiente, por ejemplo por la unidad 312 de procesamiento de transformada inversa en el decodificador 30 de video) y factores de escalado correspondientes para la transformada directa, por ejemplo por la unidad 206 de procesamiento de transformada, en un codificador 20 pueden ser especificados por consiguiente.

50

Realizaciones del codificador 20 de video (respectivamente la unidad 206 de procesamiento de transformada) pueden ser configuradas para emitir parámetros de transformada, por ejemplo un tipo de transformada o transformadas, por ejemplo directamente o codificadas o comprimidas a través de la unidad 270 de codificación entrópica, de manera que, por ejemplo, el decodificador 30 de video puede recibir y utilizar los parámetros de transformada para decodificación.

55

Cuantificación

60

La unidad 208 de cuantificación puede ser configurada para cuantificar los coeficientes 207 de transformada con el fin de obtener coeficientes cuantificados 209, por ejemplo aplicando una cuantificación escalar o cuantificación vectorial. Los coeficientes cuantificados 209 también pueden ser denominados como coeficientes 209 de transformada cuantificados o coeficientes residuales cuantificados 209.

65

El proceso de cuantificación puede reducir la profundidad de bit asociada con algunos o todos los coeficientes

207 de transformada. Por ejemplo, un coeficiente de transformada de n-bits puede ser redondeado hacia abajo a un coeficiente de transformada de m-bits durante la cuantificación, donde n es mayor que m. El grado de cuantificación puede ser modificado ajustando un parámetro de cuantificación (QP). Por ejemplo para cuantificación escalar, se puede aplicar un escalado diferente para lograr una cuantificación más fina o más aproximada. Tamaños de etapa de cuantificación más pequeños corresponden a cuantificación más fina, mientras que tamaños de etapa de cuantificación más grandes corresponden a cuantificación más aproximada. El tamaño de etapa de cuantificación aplicable puede ser indicado por un parámetro de cuantificación (QP). El parámetro de cuantificación puede ser, por ejemplo, un índice a un conjunto predefinido de tamaños de etapas de cuantificación aplicables. Por ejemplo, parámetros de cuantificación pequeños pueden corresponder a cuantificación fina (tamaños pequeños de etapas de cuantificación) y parámetros de cuantificación grandes pueden corresponder a cuantificación más aproximada (tamaños grandes de etapas de cuantificación) o viceversa. La cuantificación puede incluir división por un tamaño de etapa de cuantificación y una descuantificación inversa y/o correspondiente, por ejemplo por la unidad 210 de cuantificación inversa, puede incluir multiplicación por el tamaño de etapa de cuantificación. Realizaciones según algunas normas, por ejemplo HEVC, pueden ser configuradas para utilizar un parámetro de cuantificación con el fin de determinar el tamaño de etapa de cuantificación. Generalmente, el tamaño de etapa de cuantificación puede ser calculado basándose en un parámetro de cuantificación utilizando una aproximación de punto fijo de una ecuación que incluye división. Factores de escalado adicionales pueden ser introducidos para cuantificación y descuantificación con el fin de restaurar la norma del bloque residual que podría modificarse debido al escalado utilizado en la aproximación de punto fijo de la ecuación para tamaño de etapa de cuantificación y parámetro de cuantificación. En una implementación ejemplar, el escalado de la transformada inversa y la descuantificación podrían combinarse. Alternativamente, tablas de cuantificación personalizadas pueden ser utilizadas y señalizadas desde un codificador a un decodificador, por ejemplo en una corriente de bits. La cuantificación es una operación con pérdida, en donde la pérdida aumenta con el incremento de los tamaños de etapa de cuantificación.

Realizaciones del codificador 20 de video (respectivamente, unidad 208 de cuantificación) pueden ser configuradas para emitir parámetros de cuantificación (QP), por ejemplo directamente o codificados a través de la unidad 270 de codificación entrópica, de manera que, por ejemplo, el decodificador 30 de video puede recibir y aplicar los parámetros de cuantificación para decodificación.

Cuantificación inversa

La unidad 210 de cuantificación inversa está configurada para aplicar la cuantificación inversa de la unidad 208 de cuantificación sobre los coeficientes cuantificados para obtener coeficientes descuantificados 211, por ejemplo aplicando la inversa del esquema de cuantificación aplicado por la unidad 208 de cuantificación basándose en o utilizando el mismo tamaño de paso de cuantificación que la unidad 208 de cuantificación. Los coeficientes descuantificados 211 también pueden ser denominados como coeficientes residuales descuantificados 211 y corresponden – aunque típicamente no son idénticos a los coeficientes de transformada debido a la pérdida por cuantificación – a los coeficientes 207 de transformada.

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, por ejemplo una transformada inversa de coseno discreta (DCT) o transformada inversa de seno discreta (DST), u otras transformadas inversas para obtener un bloque residual reconstruido 213 (o coeficientes descuantificados correspondientes 213) en el dominio de muestra. El bloque residual reconstruido 213 también se puede denominar como bloque 213 de transformada.

Reconstrucción

La unidad 214 de reconstrucción (por ejemplo, sumadora 214) está configurada para añadir el bloque 213 de transformada (es decir bloque residual reconstruido 213) al bloque 265 de predicción para obtener un bloque reconstruido 215 en el dominio de muestra, por ejemplo añadiendo, muestra por muestra, los valores de muestra del bloque residual reconstruido 213 y los valores de muestra del bloque 265 de predicción.

Filtración

La unidad 220 de filtro de bucle (o para abreviar “filtro de bucle” 220), está configurada para filtrar el bloque reconstruido 215 para obtener un bloque filtrado 221, o en general, para filtrar muestras reconstruidas para obtener muestras filtradas. La unidad de filtro de bucle, por ejemplo, está configurada para suavizar las transiciones de pixel, o de otra manera mejorar la calidad del video. La unidad 220 de filtro de bucle puede comprender uno o más filtros de bucle tal como un filtro de desbloqueo, un filtro de desplazamiento adaptable a la muestra (SAO) o uno o más filtros diferentes, por ejemplo un filtro bilateral, un filtro de bucle adaptable (ALF), un filtro de afinamiento, filtros de suavización o filtros colaborativos, o cualquier combinación de los

misimos. Aunque la unidad 220 de filtro de bucle se muestra en la figura 2 como siendo un filtro de bucle, en otras configuraciones, la unidad 220 de filtro de bucle puede ser implementada como un filtro post-bucle. El bloque filtrado 221 también se puede denominar como bloque reconstruido filtrado 221.

- 5 Realizaciones del codificador 20 de video (respectivamente unidad 220 de filtro de bucle) pueden ser configuradas para emitir parámetros de filtro de bucle (tal como información de desplazamiento adaptable a la muestra), por ejemplo directamente o codificados a través de la unidad 270 de codificación entrópica, de manera que, por ejemplo, un decodificador 30 puede recibir y aplicar los mismos parámetros de filtro de bucle o filtros de bucle respectivos para decodificación.

10

Memoria tampón de imágenes decodificadas

- La memoria tampón 230 de imágenes decodificadas (DPB) puede ser una memoria que almacena imágenes de referencia o en general datos de imágenes de referencia para codificar datos de video a través del codificador 20 de video. La DPB 230 puede estar formada por cualquiera de una variedad de dispositivos de memoria, tales como memoria de acceso aleatorio dinámica (DRAM), incluyendo DRAM sincronizada (SDRAM), RAM magneto-resistiva (MRAM), RAM resistiva (RRAM), u otros tipos de dispositivos de memoria. La memoria tampón 230 de imágenes decodificadas (DPB) puede ser configurada para almacenar uno o más bloques filtrados 221. La memoria tampón 230 de imágenes decodificadas además puede ser configurada para almacenar otros bloques previamente filtrados, por ejemplo, bloques 221 previamente reconstruidos y filtrados, de la misma imagen actual o de imágenes diferentes, por ejemplo imágenes previamente reconstruidas, y puede proporcionar imágenes completas previamente reconstruidas, es decir decodificadas, (y bloques de referencia y muestras correspondientes) y/o una imagen actual parcialmente reconstruida (y bloques de referencia y muestras correspondientes), por ejemplo para inter-predicción. La memoria tampón 230 de imágenes decodificadas (DPB) también puede estar configurada para almacenar uno o más bloques reconstruidos 215 no filtrados, o en general muestras reconstruidas no filtradas, por ejemplo si el bloque reconstruido 215 no es filtrado por la unidad 220 de filtro de bucle, o cualquier otra versión adicional procesada de los bloques reconstruidos o muestras.

30 Modo de selección (División en particiones y Predicción)

- La unidad 260 de selección de modo comprende una unidad 262 de división, una unidad 244 de inter-predicción y una unidad 254 de intra-predicción, y está configurada para recibir u obtener datos de imagen originales, por ejemplo un bloque original 203 (bloque actual 203 de la imagen actual 17), y datos de imagen reconstruidos por ejemplo muestras reconstruidas filtradas y/o no filtradas o bloques de la misma imagen (actual) y/o de una o una pluralidad de imágenes previamente decodificadas, por ejemplo procedentes de la memoria tampón 230 de imágenes decodificadas u otras memorias tampón (por ejemplo memoria tampón en línea, que no se muestra). Los datos de imagen reconstruidos son utilizados como datos de imagen de referencia para predicción, por ejemplo inter-predicción o intra-predicción, para obtener un bloque 265 de predicción o predictor 265.

- La unidad 260 de selección de modo puede ser configurada para determinar o seleccionar una división en particiones para un modo de predicción de bloque actual (incluyendo no división en particiones) y un modo de predicción (por ejemplo un modo de intra o inter-predicción) y generar un bloque 265 de predicción correspondiente, el cual es utilizado para el cálculo del bloque residual 205 y para la reconstrucción del bloque reconstruido 215.

- Realizaciones de la unidad 260 de selección de modo pueden ser configuradas para seleccionar el modo de división en particiones y predicción (por ejemplo de entre aquellos soportados por o disponibles para la unidad 260 de selección de modo), que proporcione la mejor coincidencia o, en otras palabras, el residual mínimo (residual mínimo significa mejor compresión para transmisión o almacenamiento), o una sobrecarga de señalización mínima (sobrecarga de señalización mínima significa mejor compresión para transmisión o almacenamiento), o lo que considere o equilibre ambos. La unidad 260 de selección de modo puede ser configurada para determinar el modo de división en particiones y predicción basándose en la optimización de la distorsión de tasa (RDO), es decir seleccionar el modo de predicción que proporcione una distorsión de tasa mínima. Términos tales como "mejor", "mínimo", "óptimo" etc., en este contexto no se refieren necesariamente a un "mejor", "mínimo", "óptimo", etc. global sino que también se pueden referir al cumplimiento de un criterio de terminación o selección tal como un valor que excede o que queda por debajo de un umbral u otras restricciones que potencialmente conducen a una "selección sub-óptima" pero reduciendo la complejidad y el tiempo de procesamiento.

- En otras palabras, la unidad 262 de división en particiones puede ser configurada para dividir el bloque 203 en particiones de bloque más pequeñas o sub-bloques (que forman bloques nuevamente), por ejemplo utilizando iterativamente división en particiones de árbol cuaternario (QT), división en particiones binaria (BT) o división en particiones de árbol triple (TT) o cualquier combinación de las mismas, y realizar, por ejemplo, la predicción para cada una de las divisiones en particiones de bloque o sub-bloques, en donde la selección de modo

comprende la selección de la estructura de árbol del bloque 203 dividido en particiones y se aplican los modos de predicción a cada una de las divisiones en particiones de bloque o sub-bloques.

- 5 En lo sucesivo, se explicará con más detalle el procesamiento de división en particiones (por ejemplo por la unidad 260 de división en particiones) y predicción (por la unidad 244 de inter-predicción y la unidad 254 de intra-predicción) realizado por un codificador 20 de video ejemplar.

División en particiones

- 10 La unidad 262 de división en particiones puede hacer particiones (o dividir) un bloque actual 203 en particiones más pequeñas, por ejemplo, bloques más pequeños de tamaño cuadrado o rectangular. Estos bloques más pequeños (que también pueden ser denominados como sub-bloques) se pueden dividir adicionalmente en particiones incluso más pequeñas. Esto también se denomina como división en particiones de árbol o división en particiones de árbol jerárquica, en donde un bloque raíz, por ejemplo al nivel 0 de árbol raíz (nivel 0 de jerarquía, profundidad 0), puede ser dividido en particiones de manera recursiva, por ejemplo puede ser dividido en particiones en dos o más bloques de un siguiente nivel de árbol inferior, por ejemplo nodos en el nivel 1 de árbol (nivel 1 de jerarquía, profundidad 1), en donde estos bloques nuevamente pueden ser divididos en particiones en dos o más bloques de un siguiente nivel inferior, por ejemplo nivel 2 de árbol (nivel 2 de jerarquía, profundidad 2), etc., hasta que termina la división en particiones, por ejemplo debido a que se cumple con un criterio de terminación, por ejemplo se alcanza una profundidad de árbol máxima o tamaño de bloque mínimo. Bloques que no son divididos en particiones adicionalmente también son denominados como bloques-hoja o nodos de hoja del árbol. Un árbol que utiliza la división en dos particiones se denomina como árbol binario (BT), un árbol que utiliza la división en tres particiones se denomina como árbol ternario (TT), y un árbol que utiliza la división en cuatro particiones se denomina como árbol cuaternario (QT).

- 25 Tal como se mencionó antes, el término "bloque" como se utiliza en la presente memoria, puede ser una porción, en particular una porción cuadrada o rectangular, de una imagen. Con referencia, por ejemplo, a HEVC y VVC, el bloque puede ser o corresponder a una unidad de árbol de codificación (CTU), a una unidad de codificación (CU), a una unidad de predicción (PU), y a una unidad de transformada (TU) y/o a los bloques correspondientes, por ejemplo, un bloque de árbol de codificación (CTB), un bloque de codificación (CB), un bloque de transformada (TB) o bloque de predicción (PB).

- 35 Por ejemplo una unidad de árbol de codificación (CTU) puede ser o comprender un CTB de muestras de luma, dos CTB correspondientes de muestras de croma de una imagen que tiene tres agrupaciones de muestras, o un CTB de muestras de una imagen monocromática o una imagen que es codificada utilizando tres planos de color separados y estructuras de sintaxis utilizadas para codificar las muestras. De manera correspondiente, un bloque de árbol de codificación (CTB) puede ser un bloque de NxN muestras para cierto valor de N de manera que la división de un componente en CTB es una división en particiones. 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 agrupaciones de muestras, o un bloque de codificación de muestras de una imagen monocromática o una imagen que es codificada utilizando tres planos de color separados y estructuras de sintaxis utilizadas para codificar las muestras. De manera correspondiente, un bloque de codificación (CB) puede ser un bloque de MxN muestras para algunos valores de M y N de manera que la división de un CTB en bloques de codificación es una división en particiones.

- 45 En realizaciones, por ejemplo, según HEVC, una unidad de árbol de codificación (CTU) puede ser dividida en CU utilizando una estructura de árbol cuaternario indicada como árbol de codificación. La decisión respecto a si se codifica un área de imagen utilizando predicción de inter-imagen (temporal) o de intra-imagen (espacial) se realiza al nivel de CU. Cada CU puede ser dividida, 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 es transmitida al decodificador sobre una base de PU. Después de obtener el bloque residual aplicando el proceso de predicción basándose en el tipo de división de PU, una CU puede ser dividida en particiones en unidades de transformada (TU) según otra estructura de árbol cuaternario similar al árbol de codificación para la CU.

- 55 En realizaciones, por ejemplo, según la última norma de codificación de video actualmente en desarrollo, la cual se denomina como Codificación de Video Versátil (VVC), se utiliza, por ejemplo, una división en particiones de árbol cuaternario y árbol binario (QTBT) combinados, por ejemplo, para dividir en particiones un bloque de codificación. En la estructura de bloque QTBT, una CU puede tener una forma cuadrada o rectangular. Por ejemplo, una unidad de árbol de codificación (CTU) es dividida en particiones en primer lugar por una estructura de árbol cuaternario. Los nodos hoja de árbol cuaternario son divididos en particiones adicionalmente por una estructura de árbol binario o árbol ternario (o triple). Los nodos hoja de árbol de división en particiones son denominados unidades de codificación (CU), y esa segmentación es utilizada para procesamiento de predicción y transformada sin partición adicional. Esto significa que la CU, PU y TU tienen el mismo tamaño de bloque en la estructura de bloque de codificación QTBT. En paralelo, la partición múltiple, por ejemplo, la partición de árbol triple para utilizarse junto con la estructura de bloque QTBT.

En un ejemplo, la unidad 260 de selección de modo del codificador 20 de video se puede configurar para realizar cualquier combinación de las técnicas de partición aquí descritas.

5 Tal como se describió antes, el codificador 20 de video está configurado para determinar o seleccionar el mejor modo de predicción o un modo de predicción óptimo a partir de un conjunto de modos de predicción (por ejemplo, predeterminados). El conjunto de modos de predicción puede comprender, por ejemplo, modos de intra-predicción y/o modos de inter-predicción.

10 Intra-predicción

10 El conjunto de modos de intra-predicción puede comprender 35 modos de intra-predicción diferentes, por ejemplo modos no direccionales tales como el modo DC (o medio) y el modo plano o modos direccionales, por ejemplo conforme a lo definido en HEVC, o puede comprender 67 modos de intra-predicción diferentes, por ejemplo modos no direccionales tales como el modo DC (o medio) y modo plano, o modos direccionales, por ejemplo conforme a lo definido para VVC.

15 La unidad 254 de intra-predicción está configurada para utilizar muestras reconstruidas de bloques contiguos de la misma imagen actual para generar un bloque 265 de intra-predicción según un modo de intra-predicción del conjunto de modos de intra-predicción.

20 La unidad 254 de intra-predicción (o en general la unidad 260 de selección de modo) está configurada adicionalmente para emitir parámetros de intra-predicción (o en general información indicativa del modo de intra-predicción seleccionado para el bloque) a la unidad 270 de codificación entrópica en forma de elementos 266 de sintaxis para inclusión en los datos 21 de imagen codificados, de manera que, por ejemplo, el 25 decodificador 30 de video puede recibir y utilizar los parámetros de predicción para decodificación.

Inter-predicción

30 El conjunto de (o posibles) modos de inter-predicción depende de las imágenes de referencia disponibles (es decir imágenes previas al menos parcialmente decodificadas, por ejemplo almacenadas en la DBP 230) y otros parámetros de inter-predicción, por ejemplo ya sea toda la imagen de referencia o únicamente una parte, por ejemplo un área de ventana de búsqueda alrededor del área del bloque actual, de la imagen de referencia, es utilizada para buscar un mejor bloque de referencia de coincidencia, y/o por ejemplo si la interpolación de pixel es aplicada, por ejemplo interpolación de medio/semi-pel, y/o de cuarto-pel, o no.

35 De manera adicional a los modos de predicción anteriores, se puede aplicar el modo de salto, y/o modo directo.

40 La unidad 244 de inter-predicción puede incluir una unidad de estimación de movimiento (ME) y una unidad de compensación de movimiento (MC) (ninguna de ambas se muestra en la figura 2). La unidad de estimación de movimiento puede ser configurada para recibir u obtener el bloque 203 de imagen (bloque 203 de imagen actual de la imagen actual 17) y una imagen decodificada 231, o al menos uno o una pluralidad de bloques previamente reconstruidos, por ejemplo bloques reconstruidos de una o una pluralidad de otras/diferentes imágenes 231 previamente decodificadas, para estimación de movimiento. Por ejemplo, una secuencia de video puede comprender la imagen actual y las imágenes 231 previamente decodificadas, o en otras palabras, 45 la imagen actual y las imágenes 231 previamente decodificadas pueden ser parte de o formar una secuencia de imágenes que forman una secuencia de video.

50 El codificador 20, por ejemplo, puede ser configurado para seleccionar un bloque de referencia a partir de una pluralidad de bloques de referencia de las mismas o diferentes imágenes de la pluralidad de otras imágenes y proporcionar una imagen de referencia (o índice de imagen de referencia) y/o un desplazamiento (desplazamiento espacial) entre la posición (coordenadas x, y) del bloque de referencia y la posición del bloque actual como parámetros de inter-predicción a la unidad de estimación de movimiento. Este desplazamiento también se denomina vector de movimiento (MV).

55 La unidad de compensación de movimiento está configurada para obtener, por ejemplo recibir, un parámetro de inter-predicción y ejecutar inter-predicción basándose en o utilizando el parámetro de inter-predicción para obtener un bloque 265 de inter-predicción. La compensación de movimiento, realizada por la unidad de compensación de movimiento puede involucrar la extracción o generación del bloque de predicción basándose en el vector de movimiento/bloque determinado por la estimación de movimiento, posiblemente realizando 60 interpolaciones a precisión de sub-pixel. La filtración de interpolación puede generar muestras de pixel adicionales a partir de muestras de pixel conocidas, incrementando así potencialmente el número de bloques de predicción candidatos que pueden ser utilizados para codificar un bloque de imagen. Después de recibir el vector de movimiento para la PU del bloque de imagen actual, la unidad de compensación de movimiento puede ubicar el bloque de predicción al cual apunta el vector de movimiento en una de las listas de imágenes 65 de referencia.

La unidad de compensación de movimiento también puede generar elementos de sintaxis asociados con los bloques y segmentos de video para su uso por el decodificador 30 de video en la decodificación de los bloques de imagen del segmento de video. Además o como una alternativa a segmentos y elementos de sintaxis respectivos, grupos de mosaicos y/o mosaicos y elementos de sintaxis respectivos pueden ser generados o utilizados.

Codificación entrópica

La unidad 270 de codificación entrópica está configurada para aplicar, por ejemplo, un algoritmo o esquema de codificación entrópica (por ejemplo un esquema de codificación de longitud variable (VLC), un esquema VLC adaptable al contexto (CAVLC), un esquema de codificación aritmética, una binarización, una codificación aritmética binaria adaptable al contexto (CABAC), una codificación aritmética binaria adaptable al contexto basada en sintaxis (SBAC), una codificación entrópica de división en particiones del intervalo de probabilidad (PIPE) u otra metodología o técnica de codificación entrópica) o derivación (no compresión) en los coeficientes cuantificados 209, parámetros de inter-predicción, parámetros de intra-predicción, parámetros de filtro de bucle y/u otros elementos de sintaxis para obtener datos 21 de imágenes codificadas que pueden ser emitidas a través de la salida 272, por ejemplo en forma de una corriente 21 de bits codificada de manera que, por ejemplo, el decodificador 30 de video puede recibir y utilizar los parámetros para decodificación. La corriente 21 de bits codificada puede ser transmitida al decodificador 30 de video, o almacenada en una memoria para posterior transmisión o recuperación por el decodificador 30 de video.

Otras variaciones estructurales del codificador 20 de video pueden ser utilizadas para codificar la corriente de video. Por ejemplo un codificador 20 basado en no-transformada puede cuantificar la señal residual directamente sin la unidad 206 de procesamiento de transformada para ciertos bloques o fotogramas. 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 ejemplo de un decodificador 30 de video que está configurado para implementar las técnicas de la presente solicitud. El decodificador 30 de video está configurado para recibir los datos 21 de imagen codificados (por ejemplo corriente 21 de bits codificada), por ejemplo codificados por el codificador 20, para obtener una imagen decodificada 331. Los datos de imagen o corriente de bits codificados comprenden información para decodificar los datos de imagen codificados, por ejemplo, datos que representan bloques de imagen de un segmento de video codificado (y/o grupos de mosaicos o 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 (por ejemplo una sumadora 314), un filtro 320 de bucle, una memoria tampón 330 de imágenes decodificadas (DBP), una unidad 360 de aplicación de modo, una unidad 344 de inter-predicción y una unidad 354 de intra-predicción. La unidad 344 de inter-predicción puede ser o incluir una unidad de compensación de movimiento. El decodificador 30 de video, en algunos ejemplos, puede realizar una etapa de decodificación generalmente recíproca a la etapa de codificación descrita con respecto al codificador 100 de video de la figura 2.

Tal como se explicó con respecto al codificador 20, 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 tampón 230 de imágenes decodificadas (DPB), la unidad 344 de inter-predicción y la unidad 354 de intra-predicción también son denominadas como que forman el "decodificador integrado" del codificador 20 de video. 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 tampón 330 de imágenes decodificadas puede ser idéntica en función a la memoria tampón 230 de imágenes decodificadas. Por lo tanto, las explicaciones proporcionadas para las unidades y funciones respectivas del codificador 20 de video se aplican de manera correspondiente a las unidades y funciones respectivas del decodificador 30 de video.

Decodificación entrópica

La unidad 304 de decodificación entrópica está configurada para analizar la corriente 21 de bits (o en general los datos 21 de imagen codificados) y realizar, por ejemplo, decodificación entrópica a los datos 21 de imagen codificados para obtener, por ejemplo, coeficientes cuantificados 309 y/o parámetros de codificación decodificados (que no se muestran en la figura 3), por ejemplo cualquiera o todos los parámetros de inter-predicción (por ejemplo, índice de imagen de referencia y vector de movimiento), parámetros de intra-

predicción (por ejemplo modo o índice de intra-predicción), parámetros de transformada, parámetros de cuantificación, parámetros de filtro de bucle, y/u otros elementos de sintaxis. La unidad 304 de decodificación entrópica puede ser configurada para aplicar los algoritmos o esquemas de decodificación correspondientes a los esquemas de codificación tal como se describió con referencia a la unidad 270 de codificación entrópica del codificador 20. La unidad 304 de decodificación entrópica además se puede configurar para proporcionar parámetros de inter-predicción, parámetros de intra-predicción 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 video puede recibir los elementos de sintaxis al nivel de segmento de video y/o al nivel de bloque de video. Además o como una alternativa a segmentos y elementos de sintaxis respectivos, grupos de mosaicos y/o mosaicos y elementos de sintaxis respectivos pueden ser recibidos y/o utilizados.

Cuantificación inversa

La unidad 310 de cuantificación inversa puede ser configurada para recibir parámetros de cuantificación (QP) (o en general información relacionada con la cuantificación inversa) y coeficientes cuantificados procedentes de los datos 21 de imagen codificados (por ejemplo mediante análisis y/o decodificación, por ejemplo por la unidad 304 de decodificación entrópica) y aplicar, basándose en los parámetros de cuantificación, una cuantificación inversa en los coeficientes cuantificados decodificados 309 para obtener coeficientes descuantificados 311, que también se pueden denominar como coeficientes 311 de transformada. El proceso de cuantificación inversa puede incluir el uso de un parámetro de cuantificación determinado por el codificador 20 de video para cada bloque de video en el segmento de video (o mosaico o grupo de mosaicos) para determinar un grado de cuantificación y, de igual manera, un grado de cuantificación inversa que debiera aplicarse.

Transformada inversa

La unidad 312 de procesamiento de transformada inversa puede ser configurada para recibir coeficientes descuantificados 311, también denominados como coeficientes 311 de transformada, y aplicar una transformada a los coeficientes descuantificados 311 para obtener bloques residuales reconstruidos 213 en el dominio de muestra. Los bloques residuales reconstruidos 213 también se pueden denominar como bloques 313 de transformada. La transformada puede ser una transformada inversa, por ejemplo, una DCT inversa, una DST inversa, una transformada de número entero inversa, o un proceso de transformada inversa conceptualmente similar. La unidad 312 de procesamiento de transformada inversa además se puede configurar para recibir parámetros de transformada o información correspondiente de los datos 21 de imagen codificados (por ejemplo mediante análisis y/o decodificación, por ejemplo por la unidad 304 de decodificación entrópica) para determinar la transformada que se ha de aplicar a los coeficientes descuantificados 311.

Reconstrucción

La unidad 314 de reconstrucción (por ejemplo sumadora 314) puede ser configurada para añadir el bloque residual reconstruido 313 al bloque 365 de predicción para obtener un bloque reconstruido 315 en el dominio de muestra, por ejemplo añadiendo los valores de muestra del bloque residual reconstruido 313 y los valores de muestra del bloque 365 de predicción.

Filtración

La unidad 320 de filtro de bucle (ya sea en el bucle de codificación o después del bucle de codificación) está configurada para filtrar el bloque reconstruido 315 para obtener un bloque filtrado 321, por ejemplo para suavizar las transiciones de pixel, o de otra manera mejorar la calidad del video. La unidad 320 de filtro de bucle puede comprender uno o más filtros de bucle tal como un filtro de desbloqueo, un filtro de desplazamiento adaptable a la muestra (SAO) o uno o más filtros diferentes, por ejemplo, un filtro bilateral, un filtro de bucle adaptable (ALF), un filtro de afilado, un filtro de suavizado o filtros colaborativos, o cualquier combinación de los mismos. Aunque la unidad 320 de filtro de bucle se muestra en la figura 3 como siendo un filtro de bucle, en otras configuraciones, la unidad 320 de filtro de bucle puede ser implementada como un filtro post-bucle.

Memoria Tampón de Imágenes Decodificadas

Los bloques 321 de video decodificados de una imagen son luego almacenados en la memoria tampón 330 de imágenes decodificadas, la cual almacena las imágenes decodificadas 331 como imágenes de referencia para posterior compensación de movimiento para otras imágenes y/o para emitir una visualización respectivamente.

El decodificador 30 está configurado para emitir la imagen decodificada 311, por ejemplo a través de la salida 312, para presentación o visualización a un usuario.

Predicción

La unidad 344 de inter-predicción puede ser idéntica a la unidad 244 de inter-predicción (en particular a la unidad de compensación de movimiento) y la unidad 354 de intra-predicción puede ser idéntica a la unidad 254 de inter-predicción en función, y toma decisiones de división o partición y predicción basándose en los parámetros de división y/o predicción o información respectiva recibidos desde los datos 21 de imagen codificados (por ejemplo, mediante análisis y/o decodificación, por ejemplo, por la unidad 304 de decodificación entrópica). La unidad 360 de aplicación de modo puede ser configurada para ejecutar la predicción (intra o inter-predicción) por bloque con base en imágenes reconstruidas, bloques o muestras respectivas (filtradas o no filtradas) para obtener el bloque 365 de predicción.

Cuando la porción de video es codificada como un segmento intra-codificado (I), la unidad 354 de intra-predicción de la unidad 360 de aplicación de modo está configurada para generar el bloque 365 de predicción para un bloque de imagen del segmento de video actual basándose en un modo de intra-predicción señalado y datos de bloques previamente decodificados de la imagen actual. Cuando la imagen de video es codificada como segmento inter-codificado (es decir, B, o P), la unidad 344 de inter-predicción (por ejemplo 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 video de la porción de video actual basándose en los vectores de movimiento y otros elementos de sintaxis recibidos desde la unidad 304 de decodificación entrópica. Para inter-predicción, los bloques de predicción pueden ser producidos a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El decodificador 30 de video puede construir las listas de fotogramas de referencia, Lista 0 y Lista 1, utilizando técnicas de construcción por defecto basándose en las imágenes de referencia almacenadas en DPB 330. Lo mismo o similar se puede aplicar para o por realizaciones utilizando grupos de mosaicos (por ejemplo, grupos de mosaicos de video) y/o mosaicos (por ejemplo, mosaicos de video) además o alternativamente a segmentos (por ejemplo, segmentos de video), por ejemplo, un video puede ser codificado utilizando grupos de mosaicos y/o mosaicos I, P o B.

La unidad 360 de aplicación de modo está configurada para determinar la información de predicción para un bloque de video del segmento de video actual analizando los vectores de movimiento o información relacionada y otros elementos de sintaxis, y utiliza la información de predicción para producir los bloques de predicción para que el bloque de video actual sea decodificado. Por ejemplo la unidad 360 de aplicación de modo utiliza algunos de los elementos de sintaxis recibidos para determinar un modo de predicción (por ejemplo, intra o inter-predicción) utilizado para codificar los bloques de video del segmento de video, un tipo de segmento de inter-predicción (por ejemplo, segmento B, segmento P, o 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 video inter-codificado del segmento, estatus de inter-predicción para cada bloque de video inter-codificado del segmento, y otra información para decodificar los bloques de video en el segmento de video actual. Lo mismo o similar se puede aplicar para o por realizaciones utilizando grupos de mosaicos (por ejemplo, grupos de mosaicos de video) y/o mosaicos (por ejemplo, mosaicos de video) además o alternativamente a segmentos (por ejemplo, segmentos de video), por ejemplo, un video puede ser codificado utilizando grupos de mosaicos y/o mosaicos I, P o B.

Realizaciones del decodificador 30 de video como se muestra en la figura 3 pueden ser configuradas para dividir en particiones y/o decodificar la imagen utilizando segmentos (también denominados como segmentos de video), en donde una imagen puede ser dividida en particiones o decodificada utilizando uno o más segmentos (típicamente que no se superponen), y cada segmento puede comprender uno o más bloques (por ejemplo, CTU).

Realizaciones del decodificador 30 de video como se muestra en la figura 3 pueden ser configuradas para dividir en particiones y/o decodificar la imagen utilizando grupos de mosaicos (también denominados como grupos de mosaicos de video) y/o mosaicos (también denominados como mosaicos de video), en donde una imagen puede ser dividida en particiones o decodificada utilizando uno o más grupos de mosaicos (típicamente que no se superponen), y cada grupo de mosaicos puede comprender, por ejemplo, uno o más bloques (por ejemplo, CTU) o uno o más mosaicos, en donde cada mosaico, por ejemplo, puede tener forma rectangular y puede comprender uno o más bloques (por ejemplo, CTU), por ejemplo, bloques completos o fraccionarios.

Otras variaciones del decodificador 30 de video pueden ser utilizadas para decodificar los datos 21 de imagen codificados. Por ejemplo el decodificador 30 puede producir la corriente de video de salida sin la unidad 320 de filtración en bucle. Por ejemplo un decodificador 30 basado en no-transformada puede cuantificar a la inversa la señal residual directamente sin la unidad 312 de procesamiento de transformada inversa para algunos bloques o fotogramas. En otra implementación, el decodificador 30 de video puede tener la unidad 310 de cuantificación inversa y la unidad 312 de procesamiento de transformada inversa combinadas en una sola unidad.

Debe entenderse que, en el codificador 20 y el decodificador 30, un resultado de procesamiento de una etapa actual puede ser procesado adicionalmente y después emitido a la siguiente etapa. Por ejemplo después de la filtración de interpolación, la derivación de vector de movimiento o filtración en bucle, se puede ejecutar una operación adicional, tal como Recorte o desplazamiento, sobre el resultado de procesamiento de la filtración

de interpolación, derivación de vector de movimiento o filtración en bucle.

Debe observarse que operaciones adicionales pueden ser aplicadas a los vectores de movimiento derivados del bloque actual (incluyendo, entre otros, vectores de movimiento de punto de control de modo afín, vectores de movimiento de sub-bloque en modos afín, plano, ATMVP, vectores de movimiento temporal, y así sucesivamente). Por ejemplo, el valor del vector de movimiento queda restringido a un intervalo predefinido según su bit representativo. Si el bit representativo del vector de movimiento es bitDepth, entonces el intervalo es $-2^{(\text{bitDepth}-1)} \sim 2^{(\text{bitDepth}-1)}-1$, donde "A" significa exponenciación. Por ejemplo, si bitDepth se establece igual a 16, el intervalo es $-32768 \sim 32767$; si bitDepth se establece igual a 18, el intervalo es $-131072 \sim 131071$. Por ejemplo, el valor del vector de movimiento derivado (por ejemplo, los MV de cuatro sub-bloques 4x4 dentro de un bloque 8x8) está restringido de manera que la máxima diferencia entre partes enteras de los MV de cuatro sub-bloques 4x4 no es más de N píxeles, tal como no más de 1 píxel. Aquí se proporcionan dos métodos para restringir el vector de movimiento según el bitDepth.

15 Método 1: eliminar el MSB (bit más significativo) de exceso de flujo por operaciones de flujo

$$ux = (mvx + 2^{\text{bitDepth}}) \% 2^{\text{bitDepth}} \quad (1)$$

$$mvx = (ux \geq 2^{\text{bitDepth}-1}) ? (ux - 2^{\text{bitDepth}}) : ux \quad (2)$$

$$uy = (mvy + 2^{\text{bitDepth}}) \% 2^{\text{bitDepth}} \quad (3)$$

$$mvy = (uy \geq 2^{\text{bitDepth}-1}) ? (uy - 2^{\text{bitDepth}}) : uy \quad (4)$$

25 donde mvx es un componente horizontal de un vector de movimiento de un bloque de imagen o un sub-bloque, mvy es un componente vertical de un vector de movimiento de un bloque de imagen o un sub-bloque, y ux y uy indica un valor intermedio.

Por ejemplo, si el valor de mvx es -32769, después de aplicar las fórmulas (1) y (2), el valor resultante es 32767. En el sistema informático, números decimales son almacenados como complemento de dos. El complemento de dos de -32769 es 1,0111,1111,1111,1111 (17 bits), entonces el MSB es desechado, de manera que el complemento de dos resultante es 0111,1111,1111,1111 (el número decimal es 32767), que es el mismo que el obtenido aplicando las fórmulas (1) y (2).

$$ux = (mvp_x + mvd_x + 2^{\text{bitDepth}}) \% 2^{\text{bitDepth}} \quad (5)$$

$$mvx = (ux \geq 2^{\text{bitDepth}-1}) ? (ux - 2^{\text{bitDepth}}) : ux \quad (6)$$

$$uy = (mvp_y + mvd_y + 2^{\text{bitDepth}}) \% 2^{\text{bitDepth}} \quad (7)$$

$$mvy = (uy \geq 2^{\text{bitDepth}-1}) ? (uy - 2^{\text{bitDepth}}) : uy \quad (8)$$

Las operaciones pueden ser aplicadas durante la suma de mvp y mvd, como se muestra en las fórmulas (5) a (8).

Método 2: eliminar el MSB de exceso de flujo recortando el valor

$$vx = \text{Clip3}(-2^{\text{bitDepth}-1}, 2^{\text{bitDepth}-1}-1, vx)$$

$$vy = \text{Clip3}(-2^{\text{bitDepth}-1}, 2^{\text{bitDepth}-1}-1, vy)$$

donde vx es un componente horizontal de un vector de movimiento de un bloque de imagen o un sub-bloque, vy es un componente vertical de un vector de movimiento de un bloque de imagen o de un sub-bloque; x, y, y z respectivamente corresponden a los tres valores de entrada del proceso de recorte MV, y la definición de la función Clip3 es como sigue:

$$\text{Clip3}(x, y, z) = \begin{cases} x & ; & z < x \\ y & ; & z > y \\ z & ; & \text{de otra manera} \end{cases}$$

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

El dispositivo 400 de codificación de video comprende puertos 410 de ingreso (o puertos 410 de entrada) y unidades receptoras (Rx) 420 para recibir datos; un procesador, unidad lógica, o unidad de procesamiento central (CPU) 430 para procesar los datos; unidades transmisoras (Tx) 440 y 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

5 codificación de video también puede comprender componentes óptico-a-eléctrico (OE) y componentes eléctrico-a-óptico (EO) acoplados a los puertos 410 de ingreso, las unidades receptoras 420, las unidades transmisoras 440, y los puertos 450 de egreso para el egreso o ingreso de señales ópticas o eléctricas.

10 El procesador 430 es implementado por hardware y software. El procesador 430 puede ser implementado como uno o más chips de CPU, núcleos (por ejemplo, como un procesador de múltiples núcleos), FPGA, ASIC, y DSP. El procesador 430 está en comunicación con los puertos 410 de ingreso, unidades receptoras 420, unidades transmisoras 440, puertos 450 de egreso, y memoria 460. El procesador 430 comprende un módulo 470 de codificación. El módulo 470 de codificación implementa las realizaciones expuestas antes descritas. Por ejemplo, el módulo 470 de codificación implementa, procesa, prepara o proporciona las diversas

15 operaciones de codificación. La inclusión del módulo 470 de codificación proporciona por ello una mejora sustancial a la funcionalidad del dispositivo 400 de codificación de video y efectúa una transformación del dispositivo 400 de codificación de video a un estado diferente. Alternativamente, el módulo 470 de codificación es implementado como instrucciones almacenadas en la memoria 460 y ejecutadas por el procesador 430.

20 La memoria 460 puede comprender uno o más discos, unidades de cinta, y unidades de estado sólido y puede ser utilizada como un dispositivo de almacenamiento de datos de exceso de flujo, para almacenar programas cuando dichos programas son seleccionados para su ejecución, y para almacenar instrucciones y datos que son leídos durante la ejecución de programa. La memoria 460 puede ser, por ejemplo, volátil y/o no volátil y puede ser una memoria de solo lectura (ROM), memoria de acceso aleatorio (RAM), memoria ternaria de contenido que se puede direccionar (TCAM), y/o memoria de acceso aleatorio estática (SRAM).

La figura 5 es un diagrama de bloques simplificado de un aparato 500 que puede ser utilizado como uno cualquiera o ambos del dispositivo fuente 12 y del dispositivo destino 14 de la figura 1 según una realización

30 Un procesador 502 en el aparato 500 puede ser una unidad de procesamiento central. Alternativamente, el procesador 502 puede ser cualquier otro tipo de dispositivo, o múltiples dispositivos, con la capacidad para manipular o procesar información ahora existente o desarrollada más adelante. Aunque las implementaciones descritas pueden ser puestas en práctica con un solo procesador como se muestra, por ejemplo, el procesador 502, se pueden conseguir ventajas en velocidad y eficiencia utilizando más de un procesador.

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) en una implementación. Cualquier otro tipo conveniente de dispositivo de almacenamiento puede ser utilizado como la memoria 504. La memoria 504 puede incluir un código y datos 506 a los que se tiene mediante el procesador 502 utilizando un bus 512. La memoria 504 puede además

40 incluir un sistema operativo 508 y programas 510 de aplicación, incluyendo los programas 510 de aplicación al menos un programa que permita al procesador 502 realizar los métodos aquí descritos. Por ejemplo los programas 510 de aplicación pueden incluir las aplicaciones 1 a N, las cuales además incluyen una aplicación de codificación de video que realiza los métodos aquí descritos.

45 El aparato 500 también puede incluir uno o más dispositivos de salida, tal como un dispositivo 518 de visualización. El dispositivo 518 de visualización puede ser, en un ejemplo, un dispositivo de visualización táctil que combina un dispositivo de visualización con un elemento sensible al tacto que puede operar para detectar entradas táctiles. El dispositivo 518 de visualización puede ser acoplado al procesador 502 mediante el bus

50 512.

Aunque aquí se muestra como un solo bus, el bus 512 del aparato 500 puede estar compuesto de múltiples buses. Además, el almacenamiento secundario 514 puede ser directamente acoplado a los otros componentes del aparato 500 o se puede acceder al mismo a través de una red y puede comprender una sola unidad

55 integrada tal como una tarjeta de memoria o múltiples unidades tales como múltiples tarjetas de memoria. El aparato 500 puede así ser implementado en una amplia variedad de configuraciones.

La unidad 244 de inter-predicción puede incluir la unidad de estimación de movimiento (ME) y la unidad de compensación de movimiento (MC) (no se muestran en la figura 2). La unidad de estimación de movimiento está configurada para recibir u obtener el bloque 203 de imagen (bloque 203 de imagen actual de la imagen actual 201) y una imagen decodificada 331, o al menos uno o una pluralidad de bloques previamente

60 reconstruidos, por ejemplo, bloques reconstruidos de una o una pluralidad de otras/diferentes imágenes previamente decodificadas 331, para estimación de movimiento. Por ejemplo, una secuencia de video puede comprender la imagen actual y las imágenes previamente decodificadas 331, o en otras palabras, la imagen actual y las imágenes previamente decodificadas 331 pueden ser parte de o formar una secuencia de imágenes que forman una secuencia de video. El codificador 200, por ejemplo, puede ser configurado para seleccionar

un bloque de referencia a partir de una pluralidad de bloques de referencia de las mismas imágenes o imágenes diferentes de la pluralidad de otras imágenes y proporcionar una imagen de referencia (o índice de imagen de referencia, ...) y/o un desplazamiento (desplazamiento espacial) entre la posición (coordenadas x, y) del bloque de referencia y la posición del bloque actual como parámetros de inter-predicción a la unidad de estimación de movimiento (que no se muestra en la figura 2). Este desplazamiento también se denomina vector de movimiento (MV). La fusión es una herramienta de estimación de movimiento importante utilizada en HEVC y heredada a VVC.

Para realizar la estimación de fusión, se construye una lista de candidatos de fusión, cada uno de los candidatos en la lista de candidatos comprende datos de movimiento, incluyendo estos datos de movimiento información que indica si se utilizan una o dos listas de imágenes de referencia, e incluyendo también otra información tal como un índice de referencia y un vector de movimiento para cada lista. En un ejemplo, la lista de candidatos de fusión es construida con base en los siguientes candidatos:

- a. hasta cuatro candidatos de fusión espacial que son derivados a partir de cinco bloques contiguos espaciales;
- b. un candidato de fusión temporal derivado a partir de dos bloques temporales ubicados juntos;
- c. candidatos de fusión adicionales que incluyen candidatos bi-predictivos combinados y candidatos de vector de movimiento cero.

Los primeros candidatos en la lista de candidatos de fusión son los contiguos espaciales. Hasta cuatro candidatos son insertados en la lista de fusión comprobando secuencialmente A1, B1, B0, A0 y B2, en ese orden, según la parte derecha de la figura 6.

Además de comprobar si un bloque de codificación está disponible y contiene información de movimiento, se realizan algunas comprobaciones de redundancia adicionales antes de tomar todos los datos de movimiento del bloque de codificación como un candidato de fusión. Estas comprobaciones de redundancia pueden ser divididas en dos categorías: a. evitar datos de movimiento redundantes en la lista; b. impedir la fusión de dos particiones que podrían ser expresadas por otros medios que crearían sintaxis redundante.

En un ejemplo, N es el número de candidatos de fusión espaciales, una comprobación de redundancia completa comprendería $(N \cdot (N-1))/2$ comparaciones de datos de movimiento. En caso de los cinco candidatos de fusión espaciales potenciales, se requerirían diez comparaciones de datos de movimiento para garantizar que todos los candidatos en la lista de fusión tengan diferentes datos de movimiento. Durante el desarrollo de HEVC, las comprobaciones para datos de movimiento redundantes se han reducido a un subconjunto de manera en que se mantiene la eficiencia de la codificación, mientras que la lógica de comparación es reducida significativamente. Al final, no más de dos comparaciones son realizadas por candidato, teniendo como resultado cinco comparaciones generales. Dado el orden de {A1, B1, B0, A0, B2}, B0 solamente comprueba B1, A0 solamente comprueba A1, y B2 solamente comprueba A1 y B1. En un ejemplo de la comprobación de redundancia de particiones, la PU inferior de una división en particiones $2N \times N$ es fusionada con la superior eligiendo B1 candidato. Esto tendría como resultado una CU con dos PU que tienen los mismos datos de movimiento, los cuales podrían ser igualmente señalizados como una CU $2N \times 2N$. En general, esta comprobación se aplica para todas las segundas PU de las particiones rectangular y asimétrica $2N \times N$, $2N \times nU$, $2N \times nD$, $N \times 2N$, $nR \times 2N$ y $nL \times 2N$. Se observa que para los candidatos de fusión espaciales, solamente se realizan las comprobaciones de redundancia y los datos de movimiento son copiados a partir de los bloques candidatos. Por lo tanto, aquí no se requiere escalado de vector de movimiento. La derivación de los vectores de movimiento para el candidato de fusión temporal es la misma que para la Predicción de Vector de Movimiento Temporal, TMVP. Dado que un candidato de fusión comprende todos los datos de movimiento y la TMVP es solamente un vector de movimiento (eso significa, que en fusión regular solamente está permitido un candidato TMVP máximo), la derivación de los datos de movimiento depende del tipo de segmento. Para segmentos bi-predictivos, una TMVP es derivada para cada lista de imágenes de referencia. Dependiendo de la disponibilidad de la TMVP para cada lista, el tipo de predicción se establece en bi-predicción o en la lista para la cual está disponible la TMVP. Todos los índices de imagen de referencia asociados se establecen igual a cero. Para segmentos uni-predictivos, la TMVP para la lista 0 es derivada junto con el índice de imagen de referencia igual a cero.

Cuando al menos una TMVP está disponible y el candidato de fusión temporal es añadido a la lista, no se realiza una comprobación de redundancia. Esto hace que la construcción de la lista de fusión sea independiente de la imagen ubicada junto a ella lo que mejora la resiliencia de error. Considérese el caso donde el candidato de fusión temporal sería redundante y, por lo tanto, no estaría incluido en la lista de candidatos de fusión. En el caso de una imagen perdida ubicada junto a ella, el decodificador no podría derivar los candidatos temporales y, por lo tanto, no puede comprobar si serían redundantes. El indexado de todos los candidatos subsiguientes se vería afectado por esto.

Por razones de robustez del análisis, el número de candidatos en la lista de candidatos de fusión es fijo.

Después que los candidatos de fusión espaciales y temporales hayan sido añadidos, la lista puede no estar completa (un número de candidatos en la lista de candidatos de fusión es más pequeño que el número fijo). A fin de compensar la pérdida de eficiencia de la codificación que surge junto con la señalización del índice de lista adaptable de no-longitud, se generan candidatos adicionales. Dependiendo del tipo de segmento, hasta dos tipos de candidatos son utilizados para completar la lista: a. Candidatos bi-predictivos combinados; b. Candidatos de vector de movimiento cero.

En segmentos bi-predictivos, se pueden generar candidatos adicionales con base en los candidatos existentes, combinando datos de movimiento de la lista 0 de imágenes de referencia de un candidato con los datos de movimiento de la lista 1 de otro. Esto se efectúa copiando Δx_0 , Δy_0 , Δt_0 procedente de un candidato, por ejemplo el primero, y Δx_1 , Δy_1 , Δt_1 procedente de otro, por ejemplo el segundo. Las diferentes combinaciones son predefinidas y proporcionadas en la Tabla 1.

Tabla 1

Orden de combinación	0	1	2	3	4	5	6	7	8	9	10	11
$\Delta x_0, \Delta y_0, \Delta t_0$ de Cand.	0	1	0	2	1	2	0	3	1	3	2	3
$\Delta x_1, \Delta y_1, \Delta t_1$ de Cand.	1	0	2	0	2	1	3	0	3	1	3	2

Cuando la lista sigue sin estar completa después de añadir los candidatos bi-predictivos combinados, o para segmentos uni-predictivos, se añaden candidatos de vector de movimiento cero. Los candidatos de vector de movimiento cero tienen un vector de movimiento de desplazamiento cero para segmentos uni-predictivos y dos vectores de movimiento de desplazamiento cero para segmentos bi-predictivos. Los índices de referencia se establecen iguales a cero, y se incrementan por uno para cada candidato adicional, hasta que se alcanza el número máximo de índices de referencia. Si aún se necesitan añadir candidatos adicionales, se utiliza un índice de referencia igual a cero para crear candidatos adicionales. Para los candidatos adicionales, no se realizan comprobaciones de redundancia ya que la omisión de estas comprobaciones no introducirá una pérdida de eficiencia de la codificación.

Para cada PU codificada en modo de predicción inter-imagen, se utiliza un `merge_flag` ("indicador de fusión") para indicar que la fusión de bloques se utiliza para derivar los datos de movimiento o no. Se utiliza un `merge_idx` para determinar el candidato en la lista de fusión que proporciona los datos de movimiento necesarios para la predicción de movimiento compensado, MCP. Además de la señalización a nivel de PU, un número de candidatos en la lista de fusión es señalizado en la cabecera de segmento. En un ejemplo, el valor por defecto es cinco, es representado como una diferencia a cinco (`five_minus_max_num_merge_cand`). Con referencia al proceso de construcción de la lista de candidatos de fusión, el proceso general permanece igual aunque termina después que la lista contiene el número máximo de candidatos de fusión. En el diseño inicial, el valor máximo para la codificación de índice de fusión fue proporcionado por un número de candidatos espaciales y temporales disponibles en la lista. Cuando solamente hay dos candidatos disponibles, el índice puede ser codificado de manera eficiente como un indicador. A fin de analizar el índice de fusión, toda la lista de candidatos de fusión tiene que ser construida para conocer el número real de candidatos.

Una aplicación de la fusión de bloque en HEVC se combina con un modo de salto. El modo de salto es utilizado para que un bloque indique qué datos de movimiento para el bloque son inferidos, en lugar de ser señalizados explícitamente en una corriente de bits y que el residual de predicción para el bloque es cero, es decir, no se transmiten coeficientes de transformada. En HEVC, al inicio de cada CU en un segmento de predicción de inter-imagen, se señala un `skip_flag` ("indicador de salto") que implica lo siguiente: a. la CU solamente contiene una PU (tipo de partición $2N \times 2N$); b. el modo de fusión es utilizado para derivar los datos de movimiento (indicador de fusión igual a 1); c. no hay datos residuales presentes en la corriente de bits.

Otra herramienta de estimación de movimiento introducida en HEVC e inherente en VVC es denominada Predicción Avanzada de Vector de Movimiento (AMVP). En el modo AMVP, los vectores de movimiento son codificados en términos de componentes horizontal (x) y vertical (y) como una diferencia a un denominado predictor de movimiento (MVP). El cálculo de los componentes de diferencia de vector de movimiento (MVD) se muestra como $MVD_x = MV_x - MVP_x$, $MVD_y = MV_y - MVP_y$.

Vectores de movimiento de un bloque actual generalmente son correlacionados con los vectores de movimiento de bloques contiguos en la imagen actual o en las imágenes codificadas previas. Debido a que es muy probable que los bloques contiguos correspondan al mismo objeto en movimiento con movimiento similar, y no es probable que el movimiento del objeto cambie bruscamente con el paso del tiempo. En consecuencia, el uso de los vectores de movimiento en bloques contiguos como predictores reduce el tamaño de la diferencia señalizada de vector de movimiento. Las MVP generalmente son derivadas de vectores de movimiento decodificados de bloques contiguos espaciales o de bloques contiguos temporales en la imagen ubicada junto a ella. En HEVC, el enfoque de derivar implícitamente la MVP fue reemplazado por una técnica conocida como competencia de vector de movimiento, la cual señala explícitamente la MVP de una lista de MVP, y se utiliza

para derivación de vector de movimiento. La estructura de bloque de árbol cuaternario de codificación variable en HEVC puede dar como resultado un bloque que tenga varios bloques contiguos con vectores de movimiento como candidatos de MVP potenciales. Tomando como un ejemplo el contiguo a la izquierda, en el caso de que un bloque de predicción de luma 64x64 pudiera tener 16 bloques de predicción de luma 8x4 a la izquierda, cuando un bloque de árbol de codificación de luma 64x64 no es dividido adicionalmente y el de la izquierda es dividido a la profundidad máxima. La Predicción Avanzada de Vector de Movimiento (AMVP) fue introducida para modificar la competencia del vector de movimiento para tener en cuenta dicha estructura de bloque flexible. Durante el desarrollo de HEVC, el diseño inicial de AMVP fue significativamente simplificado para proporcionar una buena compensación entre la eficiencia de la codificación y un diseño de implementación amigable.

El diseño inicial de AMVP incluyó cinco MVP procedentes de tres clases diferentes de predictores: tres vectores de movimiento procedentes de contiguos espaciales, la media de los tres predictores espaciales y un vector de movimiento escalado procedente de un bloque temporalmente contiguo, ubicado junto a él. Además, la lista de predictores fue modificada reordenándola para colocar el predictor de movimiento más probable en la primera posición, y eliminando candidatos redundantes para garantizar una sobrecarga de señalización mínima. Experimentos exhaustivos a lo largo del proceso de estandarización investigaron cómo la complejidad de esta predicción de vector de movimiento, y el esquema de señalización podrían ser reducidos sin sacrificar demasiada eficiencia de la codificación. Esto conduce a simplificaciones significativas del diseño AMVP tales como eliminar el predictor medio, reducir el número de candidatos en la lista de cinco a dos, fijar el orden de los candidatos en la lista y reducir el número de comprobaciones de redundancia. El diseño de la construcción de lista de candidatos AMVP incluye los siguientes dos candidatos MVP:

- hasta dos candidatos espaciales MVP que son derivados a partir de cinco bloques contiguos espaciales;
- un candidato temporal MVP derivado de dos bloques temporales ubicados juntos cuando ambos candidatos espaciales MVP no están disponibles o son idénticos;
- vectores de movimiento cero cuando los candidatos espaciales, temporales o ambos no están disponibles.

En la descripción de candidato espacial, el flujo del proceso de derivación para los dos candidatos espaciales A y B se muestra en la figura 13. Para el candidato A, datos de movimiento procedentes de los dos bloques A0 y A1 en la esquina izquierda inferior son tomados en cuenta en un enfoque de dos pases. En el primer pase, se comprueba si cualquiera de los bloques candidatos contiene un índice de referencia que sea igual al índice de referencia del bloque actual. El primer vector de movimiento encontrado será tomado como el candidato A. Cuando todos los índices de referencia desde A0 y A1 están apuntando a una imagen de referencia diferente con el índice de referencia del bloque actual, el vector de movimiento asociado no puede ser utilizado como está. Por lo tanto, en un segundo pase, los vectores de movimiento necesitan ser escalados según las distancias temporales entre la imagen de referencia candidata y la imagen de referencia actual. La distancia temporal es expresada en términos de diferencia entre los valores de recuento de orden de imagen (POC), valores POC son utilizados para definir el orden de despliegue de las imágenes.

Para el candidato B, los candidatos B0 a B2 son comprobados en secuencia en la misma forma que A0 y A1. Sin embargo, el segundo pase es realizado cuando los bloques A0 y A1 no contienen ninguna información de movimiento, es decir, no están disponibles o codificados utilizando predicción de intra-imagen. Después, el candidato A se establece igual al candidato B no escalado, si se encuentra, y el candidato B se establece igual a una segunda variante escalada o no-escalada del candidato B. El segundo pase puede ser finalizado cuando aún podría haber candidatos no-escalados potenciales, el segundo pase busca MV no-escalados así como escalados derivados de los candidatos B0 a B2. En general, este diseño permite procesar A0 y A1 independientemente de B0, B1, y B2. La derivación de B debe tener conocimiento de la disponibilidad tanto de A0 como de A1, para buscar un MV escalado o un no-escalado adicional derivado de B0 a B2. Esta dependencia reduce significativamente las operaciones de escalado complejas del vector de movimiento para el candidato B. La reducción del número de escalados de vector de movimiento representa una reducción de complejidad significativa en el proceso de derivación del predictor de vector de movimiento.

En el proceso de selección de candidato temporal. Se puede ver a partir de la figura 6 que vectores de movimiento procedentes de bloques contiguos espaciales a la izquierda y encima del bloque actual son considerados como candidatos MVP espaciales. Esto se puede explicar ya que los bloques a la derecha y debajo del bloque actual no están decodificados todavía y por lo tanto, sus datos de movimiento no están disponibles. Dado que la imagen ubicada junto a ella es una imagen de referencia que ya está decodificada, es posible también considerar datos de movimiento, procedentes del bloque en la misma posición, procedentes de bloques a la derecha del bloque ubicado junto a él o procedentes del bloque de debajo. En HEVC, el bloque a la derecha inferior y en el centro del bloque actual han sido determinados como los más adecuados para proporcionar una buena predicción de vector de movimiento temporal (TMVP). Estos candidatos están

- ilustrados en la figura 6, donde C0 representa el bloque contiguo a la derecha debajo y C1 representa el bloque central. Primero se consideran datos de movimiento de C0, si no hay disponibles datos de movimiento de C0, se utilizan datos de movimiento del bloque de candidatos ubicado junto a él en el centro para derivar el candidato C temporal de MVP. Los datos de movimiento de C0 también son considerados como no estando disponibles cuando la PU asociada pertenece a una CTU más allá de la fila de la CTU actual. Esto minimiza los requisitos de ancho de banda de memoria para almacenar los datos de movimiento ubicados junto. En contraste con los candidatos MVP espaciales, donde los vectores de movimiento se pueden referir a la misma imagen de referencia, el escalado del vector de movimiento es obligatorio para la TMVP.
- 5 Tanto para la construcción de la lista de Fusión como para la construcción de la lista AMVP, el predictor de vector de movimiento basado en el histórico (HMVP) a la lista de fusión o AMVP. Los candidatos de fusión MVP basados en el histórico (HMVP) son añadidos a la lista de fusión/AMVP después de la MVP espacial y de la TMVP. En este método, la información de movimiento de un bloque previamente codificado es almacenada en una tabla y utilizada como MVP para la CU actual. La tabla con múltiples candidatos HMVP se mantiene durante el proceso de codificación/decodificación. La tabla es reestablecida (vacía) cuando se encuentra una nueva fila de CTU. Siempre que hay una CU inter-codificada de no-sub-bloque, la información de movimiento asociada es añadida a la última entrada de la tabla como un nuevo candidato HMVP.
- 10 En VTM4, el tamaño S de la tabla HMVP se establece para que sea 6, lo cual indica que hasta 6 candidatos MVP basados en el histórico (HMVP) pueden ser añadidos a la tabla. Cuando se inserta un nuevo candidatos de movimiento a la tabla, se utiliza una regla, primero en entrar, primero en salir (FIFO) restringida, en donde la comprobación de redundancia es aplicada en primer lugar para encontrar si existe un HMVP idéntico en la tabla. Si existe un HMVP idéntico en la tabla, el HMVP idéntico es eliminado de la tabla y todos los candidatos HMVP posteriormente son movidos hacia delante.
- 15 Los candidatos HMVP podrían ser utilizados en el proceso de construcción de la lista de candidatos de fusión/lista de AMVP. Los últimos candidatos HMVP en la tabla son comprobados en orden e insertados en la lista de candidatos después del candidato de TMVP. Se aplica comprobación de redundancia en los candidatos HMVP a los candidatos de fusión espacial o temporal.
- 20 Para reducir el número de operaciones de comprobación de redundancia, se introducen las siguientes simplificaciones:
- 25 Número de candidatos HMPV es utilizado para que la generación de lista de fusión se establezca como $(N \leq 4) ? M : (8 - N)$, en donde N indica el número de candidatos existentes en la lista de fusión y M indica el número de candidatos HMVP disponibles en la tabla. Una vez que el número total de candidatos de fusión disponibles es igual a los candidatos de fusión máximos permitidos menos 1, termina el proceso de construcción de la lista de candidatos de fusión a partir de HMVP.
- 30 En el Borrador de VVC de forma paralela a Inter modo, se introduce el modo IBC.
- 35 La copia de intra-bloque (IBC) es una herramienta adoptada en extensiones HEVC en SCC. Esto mejora significativamente la eficiencia de la codificación de materiales de contenido de pantalla. Dado que el modo IBC es implementado como un modo de codificación a nivel de bloque, la correspondencia de bloque (BM) se realiza en el codificador para encontrar el vector de bloque óptimo (o vector de movimiento) para cada CU. Aquí, un vector de movimiento es utilizado para indicar el desplazamiento del bloque actual a un bloque de referencia, el cual ya está reconstruido dentro de la imagen actual. El vector de movimiento de luma de una CU codificada por IBC tiene una precisión de número entero. El vector de movimiento de croma es acortado a la precisión de número entero también. Cuando se combina con AMVR, el modo IBC puede conmutar entre precisiones de vector de movimiento 1-pel y 4-pel. Una CU codificada por IBC es tratada como el tercer modo de predicción diferente a los modos de intra o inter-predicción.
- 40 Para reducir el consumo de memoria y la complejidad del decodificador, el IBC en VTM4 permite que se utilice la porción reconstruida del área predefinida incluyendo la CTU actual. Esta restricción permite que el modo IBC sea implementado utilizando memoria en-chip local para implementaciones de hardware.
- 45 En el lado del codificador, se ejecuta estimación de movimiento basada en función Hash para IBC. El codificador realiza comprobación RD para bloques con anchura o altura no mayor de 16 muestras de luma. Para modo de no-fusión, la búsqueda del vector de bloque es realizada utilizando en primer lugar la búsqueda basada en función Hash. Si la búsqueda basada en función hash no devuelve un candidato válido, se realizará una búsqueda local basada en correspondencia de bloques.
- 50 En la búsqueda basada en función Hash, la correspondencia de la clave Hash (CRC de 32-bits) entre el bloque actual y un bloque de referencia se extiende a todos los tamaños de bloque permitidos. El cálculo de la clave Hash para cada posición en la imagen actual se basa en sub-bloques 4x4. Para el bloque actual de un tamaño más grande, se determina una clave Hash para que corresponda con la del bloque de referencia cuando todas
- 55
- 60
- 65

las claves Hash de todos los sub-bloques 4x4 corresponden con las claves Hash en las ubicaciones de referencia correspondientes. Si se encuentra que las claves Hash de múltiples bloques de referencia corresponden con la del bloque actual, se calculan los costes del vector de bloque de cada referencia correspondiente y se selecciona aquel con el coste mínimo.

5

En la búsqueda de la correspondencia de bloque, el intervalo de búsqueda se establece para que sea N muestras a la izquierda y encima del bloque actual dentro de la CTU actual. Al inicio de una CTU, el valor de N es inicializado en 128 si no existe una imagen de referencia temporal, o es inicializado en 64 si existe al menos una imagen de referencia temporal. Una relación de acierto Hash se define como el porcentaje de muestras en la CTU que encontraron una correspondencia utilizando la búsqueda basada en Hash. Mientras se codifica la CTU actual, si la relación de acierto Hash está por debajo de 5%, N es reducido a la mitad.

10

Al nivel de CU, el modo IBC es señalizado con un indicador y puede ser señalizado como modo IBC AMVP o modo de salto/fusión IBC de la siguiente manera:

15

Modo de salto/fusión IBC: un índice de candidato de fusión es utilizado para indicar qué vectores de bloque en la lista de bloques codificados IBC candidatos contiguos son utilizados para predecir el bloque actual. La lista de fusión consiste de candidatos espaciales, HMVP, y a modo de par.

20

Modo IBC AMVP: la diferencia del vector de bloque es codificada de la misma forma que una diferencia de vector de movimiento. El método de predicción de vector de bloque utiliza dos candidatos como predictores, uno del contiguo a la izquierda y uno del contiguo por encima (si está codificado por IBC). Cuando algún contiguo no está disponible, un vector de bloque por omisión se utilizará como un predictor. Un indicador es señalizado para indicar el índice de predictor de vector de bloque.

25

Dado que IBC introdujo el modo de fusión/salto de IBC y el modo IBC AMVP, existe una lista de fusión IBC adicional y una lista AMVP que se han de construir en la propuesta VVC 4.0.

30

Dado que solamente candidatos espaciales (bloque contiguo a la izquierda A1, bloque contiguo B1 de encima, bloque contiguo A0 izquierdo inferior, bloque contiguo B0 de encima a la derecha y bloque contiguo B2 de encima a la izquierda como se muestra en la figura 6), candidatos HMVP ($H_1 \dots H_k$, k es igual al tamaño máximo de lista HMVP) y candidatos a modo de par son utilizados para construir la lista de fusión IBC en orden en el Borrador 4.0 de VVC. El siguiente recorte es posible cuando se realiza durante la construcción de la lista de fusión IBC:

35

- Recorte entre A1 y B1

- Recorte entre A0 y A1

40

- Recorte entre B0 y B1

- Recorte entre B2 y A1

- Recorte entre B2 y B1

45

- Recorte entre el último candidato H_k de HMVP y A1

- Recorte entre el último candidato H_k de HMVP y B1

50

- Recorte entre el penúltimo candidato H_{k-1} de HMVP y A1

- Recorte entre el penúltimo candidato H_{k-1} de HMVP y B1

55

El procesamiento de recorte significa comparar si dos candidatos de fusión IBC son el mismo. De manera más específica, el procesamiento de recorte compara si los vectores de bloque entre dos candidatos de fusión IBC son el mismo.

60

En resumen, el número máximo de 9 recortes es necesario para construir la lista de fusión IBC del bloque actual. En el codificador y decodificador, el recorte es retrasar el proceso de construcción de la lista de fusión. Debido a que en cada etapa del recorte, se comprueba una condición "sí", el proceso de construcción de lista de fusión adicional se realiza dependiendo de las comprobaciones de esta condición "sí". Cuanto más recorte hay en la construcción de la lista de candidatos de fusión, más complejos son los procesos del codificador y del decodificador. A fin de reducir la complejidad de la construcción de la lista de fusión, se introducen las siguientes soluciones.

65

Solución 1

Dado que los bloques contiguos espaciales de izquierda (A1) y de encima (B1) son importantes para predecir el bloque actual utilizando el modo IBC. En la solución 1, el recorte del bloque contiguo espacial entre A1 y B1 se mantiene, el recorte del bloque contiguo espacial restante se elimina. El recorte de candidato HMVP se mantiene. El siguiente recorte es posible realizado durante la construcción de la lista de fusión IBC:

- Recorte entre A1 y B1
- Recorte entre el último candidato H_k de HMVP y A1
- Recorte entre el último candidato H_k de HMVP y B1
- Recorte entre el penúltimo candidato H_{k-1} de HMVP y A1
- Recorte entre el penúltimo candidato H_{k-1} de HMVP y B1.

En este caso, el número máximo de recortes para construir la lista de candidatos de fusión IBC del bloque actual se reduce de 9 a 5. La solución redujo significativamente la complejidad de la construcción de la lista de fusión IBC tanto para el codificador como para el decodificador.

En un ejemplo, la lista de candidatos de fusión IBC concreta es construida de la siguiente manera:

Si el bloque contiguo A1 está disponible y está utilizando modo IBC, el vector de bloque del bloque A1 es insertado en la lista de candidatos de fusión IBC del bloque actual. De otra manera (A1 no está disponible o A1 no está utilizando modo IBC), el vector de bloque del bloque A1 no es insertado en la lista de candidatos de fusión IBC del bloque actual (primer candidato, sin recorte). Si el bloque contiguo B1 está disponible y está utilizando modo IBC, recortar (o determinar) si el vector de bloque de B1 es el mismo que A1. Si el vector de bloque de B1 no es el mismo que el vector de bloque de A1, el vector de bloque del bloque B1 es insertado en la lista de candidatos de fusión IBC del bloque actual. De otra manera (el vector de bloque de B1 es el mismo que el vector de bloque de A1, o B1 no está disponible o B1 no está utilizando modo IBC), el vector de bloque del bloque B1 no es insertado en la lista de candidatos de fusión IBC del bloque actual (recortar A1 y B1).

Si el bloque contiguo B0 está disponible y está utilizando modo IBC, el vector de bloque del bloque B0 es insertado en la lista de candidatos de fusión IBC del bloque actual. De otra manera (B0 no está disponible o B0 no está utilizando modo IBC), el vector de bloque del bloque B0 no es insertado en la lista de candidatos de fusión IBC del bloque actual (no recortar).

Si el bloque contiguo A0 está disponible y está utilizando modo IBC, el vector de bloque del bloque A0 es insertado en la lista de candidatos de fusión IBC del bloque actual. De otra manera (A0 no está disponible o A0 no está utilizando modo IBC), el vector de bloque del bloque A0 no es insertado en la lista de candidatos de fusión IBC del bloque actual (no recortar).

Si el bloque contiguo B2 está disponible y está utilizando modo IBC, y el tamaño de la lista de candidatos de fusión IBC actual es más pequeño que 4, el vector de bloque del bloque B2 es insertado en la lista de candidatos de fusión IBC del bloque actual. De otra manera (B2 no está disponible o B2 no está utilizando el modo IBC o el tamaño de la lista de candidatos de fusión IBC actual no es más pequeño que 4), el vector de bloque del bloque B2 no es insertado en la lista de candidatos de fusión IBC del bloque actual (no recortar).

Si el último candidato H_k de HMVP está disponible y está utilizando modo IBC, y el tamaño de lista de candidatos de fusión IBC actual es más pequeño que un valor, el valor igual al número máximo de candidatos de fusión IBC menos uno; y el vector de bloque de H_k no es el mismo que A1 y B1, insertar el vector de bloque de H_k en la lista de candidatos de fusión IBC del bloque actual. De otra manera (H_k no está disponible o H_k no está utilizando el modo IBC o el tamaño de la lista de candidatos de fusión IBC actual no es más pequeño que el número máximo de candidatos de fusión IBC menos uno, o el vector de bloque de H_k es el mismo que el vector de bloque de A1, o el vector de bloque de H_k es el mismo que el vector de bloque de B1) no insertar el vector de bloque de H_k en la lista de candidatos de fusión IBC del bloque actual (recortar H_k y A1, H_k y B1).

Si el penúltimo candidato H_{k-1} de HMVP está disponible y está utilizando modo IBC, y el tamaño de lista de candidatos de fusión IBC actual es más pequeño que un valor, el valor igual al número máximo de candidatos de fusión IBC menos uno, y el vector de bloque de H_{k-1} no es el mismo que A1 y B1, insertar el vector de bloque de H_{k-1} en la lista de candidatos de fusión IBC del bloque actual. De otra manera (H_{k-1} no está disponible o H_{k-1} no está utilizando el modo IBC o el tamaño de la lista de candidatos de fusión IBC actual no es más pequeño que el número máximo de candidatos de fusión IBC menos uno, o el vector de bloque de H_{k-1} es el mismo que el vector de bloque de A1, o el vector de bloque de H_{k-1} es el mismo que el vector de bloque de B1) no insertar

el vector de bloque de H_{k-1} en la lista de candidatos de fusión IBC del bloque actual (recortar H_{k-1} y A1, H_{k-1} y B1).

5 Insertar el resto de los candidatos HMVP uno por uno, si el candidato está disponible y está utilizando modo IBC, y el tamaño de lista de candidatos de fusión IBC actual es más pequeño que el número máximo de candidatos de fusión IBC menos uno (no recortar).

10 Insertar el candidato en modo de pares, el tamaño de lista de candidatos de fusión IBC actual es más pequeño que el número máximo de candidatos de fusión IBC (no recortar).

Solución 2

15 Dado que los bloques contiguos espaciales de izquierda (A1) y de encima (B1) son importantes para predecir el bloque actual utilizando modo IBC. En la solución 2, el bloque contiguo espacial A1 y B1 se mantiene para insertarlo en el candidato de fusión IBC, el resto de los candidatos de bloques contiguos espaciales son eliminados. El recorte de candidato HMVP se mantiene como está.

El siguiente recorte es posible cuando se realiza durante la construcción de la lista de fusión IBC:

- 20 • Recorte entre A1 y B1
- Recorte entre el último candidato H_k de HMVP y A1
- Recorte entre el último candidato H_k de HMVP y B1
- 25 • Recorte entre el penúltimo candidato H_{k-1} de HMVP y A1
- Recorte entre el penúltimo candidato H_{k-1} de HMVP y B1.

30 En este caso, el número máximo de recortes para construir la lista de candidatos de fusión IBC del bloque actual se reduce de 9 a 5. La solución redujo significativamente la complejidad de la construcción de la lista de fusión IBC, tanto para el codificador como para el decodificador.

En un ejemplo, la lista de candidatos de fusión IBC concreta es construida de la siguiente manera:

35 Si el bloque contiguo A1 está disponible y está utilizando modo IBC, insertar el vector de bloque del bloque A1 en la lista de candidatos de fusión IBC del bloque actual. De otra manera (A1 no está disponible o A1 no está utilizando modo IBC), no insertar el vector de bloque del bloque A1 en la lista de candidatos de fusión IBC del bloque actual (primer candidato, no recortar).

40 Si el bloque contiguo B1 está disponible y está utilizando modo IBC, recortar (o determinar) si el vector de bloque de B1 es el mismo que A1. Si el vector de bloque de B1 no es el mismo que el vector de bloque de A1, insertar el vector de bloque del bloque B1 en la lista de candidatos de fusión IBC del bloque actual. De otra manera (el vector de bloque de B1 es el mismo que el vector de bloque A1, o B1 no está disponible o B1 no está utilizando modo IBC), no insertar el vector de bloque del bloque A1 en la lista de candidatos de fusión IBC del bloque actual (recortar A1 y B1).

50 Si el último candidato H_k de HMVP está disponible y está utilizando modo IBC, y el tamaño de lista de candidatos de fusión IBC actual es más pequeño que un valor, el valor igual al número máximo de candidatos de fusión IBC menos uno, y el vector de bloque de H_k no es el mismo que A1 y B1, insertar el vector de bloque de H_k en la lista de candidatos de fusión IBC del bloque actual. De otra manera (H_k no está disponible o H_k no está utilizando el modo IBC o el tamaño de la lista de candidatos de fusión IBC actual no es más pequeño que el número máximo de candidatos de fusión IBC menos uno, o el vector de bloque de H_k es el mismo que el vector de bloque de A1, o el vector de bloque de H_k es el mismo que el vector de bloque de B1), no insertar el vector de bloque de H_k en la lista de candidatos de fusión IBC del bloque actual (recortar H_k y A1, H_k y B1).

60 Si el penúltimo candidato HMVP H_{k-1} está disponible y está utilizando modo IBC, y el tamaño de lista de candidatos de fusión IBC actual es más pequeño que un valor, el valor igual al número máximo de candidatos de fusión IBC menos uno, y el vector de bloque de H_{k-1} no es el mismo que A1 y B1, insertar el vector de bloque de H_{k-1} en la lista de candidatos de fusión IBC del bloque actual. De otra manera (H_{k-1} no está disponible o H_{k-1} no está utilizando el modo IBC o el tamaño de la lista de candidatos de fusión IBC actual no es más pequeño que el número máximo de candidatos de fusión IBC menos uno, o el vector de bloque de H_{k-1} es el mismo que el vector de bloque de A1, o el vector de bloque de H_{k-1} es el mismo que el vector de bloque de B1) no insertar el vector de bloque de H_{k-1} en la lista de candidatos de fusión IBC del bloque actual (recortar H_{k-1} y A1, H_{k-1} y B1).

- Insertar el resto de los candidatos HMVP uno por uno, si el candidato está disponible y está utilizando modo IBC, y el tamaño de lista de candidatos de fusión IBC actual es más pequeño que el número máximo de candidatos de fusión IBC menos uno. (no recortar). Insertar el candidato a modo de pares, el tamaño de lista de candidatos de fusión IBC actual es más pequeño que el número máximo de candidatos de fusión IBC (no recortar).

Solución 3

- Según la solución 3, el recorte de vector de bloque del bloque contiguo espacial se mantiene igual. Para el recorte de candidato HMVP, el vector de bloque del último candidato H_k de HMVP es recortado con el bloque contiguo espacial del vector de bloque A1 y B1. El resto del recorte de candidato HMVP es eliminado. En una modalidad, el siguiente recorte es posible cuando se realiza durante la construcción de la lista de fusión IBC:
- Recorte entre A1 y B1
 - Recorte entre A0 y A1
 - Recorte entre B0 y B1
 - Recorte entre B2 y A1
 - Recorte entre B2 y B1
 - Recorte entre el último candidato H_k de HMVP y A1
 - Recorte entre el último candidato H_k de HMVP y B1.

- En este caso, el número máximo de recortes para construir la lista de candidatos de fusión IBC del bloque actual se reduce de 9 a 7. La solución redujo significativamente la complejidad de la construcción de la lista de fusión IBC tanto para el codificador como para el decodificador.

En un ejemplo, la lista de candidatos de fusión IBC concreta es construida de la siguiente manera:

- Si el bloque contiguo A1 está disponible y está utilizando modo IBC, insertar el vector de bloque del bloque A1 en la lista de candidatos de fusión IBC del bloque actual. De otra manera (A1 no está disponible o A1 no está utilizando modo IBC), no insertar el vector de bloque del bloque A1 en la lista de candidatos de fusión IBC del bloque actual (primer candidato, no recortar).
- Si el bloque contiguo B1 está disponible y está utilizando modo IBC, recortar (o determinar) si el vector de bloque de B1 es el mismo que A1. Si el vector de bloque de B1 no es el mismo que el vector de bloque de A1, insertar el vector de bloque del bloque B1 en la lista de candidatos de fusión IBC del bloque actual. De otra manera (el vector de bloque de B1 es el mismo que el vector de bloque A1, o B1 no está disponible o B1 no está utilizando modo IBC), no insertar el vector de bloque del bloque A1 en la lista de candidatos de fusión IBC del bloque actual (recortar A1 y B1).
- Si el bloque contiguo B0 está disponible y está utilizando modo IBC, recortar (o determinar) si el vector de bloque de B0 es el mismo que B1. Si el vector de bloque de B0 no es el mismo que el vector de bloque de B1, insertar el vector de bloque del bloque B0 en la lista de candidatos de fusión IBC del bloque actual. De otra manera (el vector de bloque de B0 es el mismo que el vector de bloque B1, o B0 no está disponible o B0 no está utilizando modo IBC), no insertar el vector de bloque del bloque B0 en la lista de candidatos de fusión IBC del bloque actual (recortar B0 y B1).
- Si el bloque contiguo A0 está disponible y está utilizando modo IBC, recortar (o determinar) si el vector de bloque de A0 es el mismo que A1. Si el vector de bloque de A0 no es el mismo que el vector de bloque de A1, insertar el vector de bloque del bloque A0 en la lista de candidatos de fusión IBC del bloque actual. De otra manera (el vector de bloque de A0 es el mismo que el vector de bloque A1, o A0 no está disponible o A0 no está utilizando modo IBC), no insertar el vector de bloque del bloque A0 en la lista de candidatos de fusión IBC del bloque actual (recortar A0 y A1).
- Si el bloque contiguo B2 está disponible y está utilizando modo IBC, y el tamaño de lista de fusión IBC actual es más pequeño que 4, recortar (o determinar) si el vector de bloque de B2 es el mismo que A1 y si el vector de bloque de B2 es el mismo que B1. Si el vector de bloque de B2 no es el mismo que el vector de bloque de A1, y el vector de bloque de B2 no es el mismo que B1, insertar el vector de bloque del bloque B2 en la lista de candidatos de fusión IBC del bloque actual. De otra manera (el vector de bloque de B2 es el mismo que el

vector de bloque A1 o B1, o B0 no está disponible o B0 no está utilizando modo IBC), no insertar el vector de bloque del bloque B0 en la lista de candidatos de fusión IBC del bloque actual (recortar B2 y A1 y recortar B2 y B1).

- 5 Si el último candidato H_k de HMVP está disponible y está utilizando modo IBC, y el tamaño de lista de candidatos de fusión IBC actual es más pequeño que un valor, el valor igual al número máximo de candidatos de fusión IBC menos uno, y el vector de bloque de H_k no es el mismo que A1 y B1, insertar el vector de bloque de H_k en la lista de candidatos de fusión IBC del bloque actual. De otra manera (H_k no está disponible o H_k no está utilizando el modo IBC o el tamaño de la lista de candidatos de fusión IBC actual no es más pequeño que el número máximo de candidatos de fusión IBC menos uno, o el vector de bloque de H_k es el mismo que el vector de bloque de A1, o el vector de bloque de H_k es el mismo que el vector de bloque de B1) no insertar el vector de bloque de H_k en la lista de candidatos de fusión IBC del bloque actual (recortar H_k y A1, H_k y B1).

- 15 Insertar los candidatos HMVP restantes uno por uno, si el candidato está disponible y está utilizando modo IBC y el tamaño de lista de candidatos de fusión IBC actual es más pequeño que un valor, el valor igual al número máximo de candidatos de fusión IBC menos uno (no recortar).

- 20 Insertar el candidato a modo de pares, el tamaño de lista de candidatos de fusión IBC actual es más pequeño que el número máximo de candidatos de fusión IBC (no recortar).

Solución 4

- 25 Según la solución 4, el recorte de vector de bloque del bloque contiguo espacial se mantiene igual. Para el recorte de candidato HMVP, el vector de bloque del último candidato H_k de HMVP y el penúltimo candidato HMVP H_{k-1} es recortado con el bloque contiguo espacial del vector de bloque A1. El recorte de candidato HMVP restante se elimina. En una modalidad, el siguiente recorte es posible cuando se realiza durante la construcción de la lista de fusión IBC:

- 30 • Recorte entre A1 y B1
- Recorte entre A0 y A1
- Recorte entre B0 y B1
- 35 • Recorte entre B2 y A1
- Recorte entre B2 y B1
- Recorte entre el último candidato H_k de HMVP y A1
- 40 • Recorte entre el penúltimo candidato H_{k-1} de HMVP y A1.

- 45 En este caso, el número máximo de recorte para construir la lista de candidatos de fusión IBC del bloque actual se reduce de 9 a 7. La solución redujo significativamente la complejidad de la construcción de la lista de fusión IBC tanto para el codificador como para el decodificador.

En un ejemplo, la lista de candidatos de fusión IBC concreta es construida de la siguiente manera:

- 50 Si el bloque contiguo A1 está disponible y está utilizando modo IBC, insertar el vector de bloque del bloque A1 en la lista de candidatos de fusión IBC del bloque actual. De otra manera (A1 no está disponible o A1 no está utilizando modo IBC), no insertar el vector de bloque del bloque A1 en la lista de candidatos de fusión IBC del bloque actual (primer candidato, no recortar).

- 55 Si el bloque contiguo B1 está disponible y está utilizando modo IBC, recortar (o determinar) si el vector de bloque de B1 es el mismo que A1. Si el vector de bloque de B1 no es el mismo que el vector de bloque de A1, insertar el vector de bloque del bloque B1 en la lista de candidatos de fusión IBC del bloque actual. De otra manera (el vector de bloque de B1 es el mismo que el vector de bloque A1, o B1 no está disponible o B1 no está utilizando modo IBC), no insertar el vector de bloque del bloque A1 en la lista de candidatos de fusión IBC del bloque actual (recortar A1 y B1).

- 60 Si el bloque contiguo B0 está disponible y está utilizando modo IBC, recortar (o determinar) si el vector de bloque de B0 es el mismo que B1. Si el vector de bloque de B0 no es el mismo que el vector de bloque de B1, insertar el vector de bloque del bloque B0 en la lista de candidatos de fusión IBC del bloque actual. De otra manera (el vector de bloque de B0 es el mismo que el vector de bloque B1, o B0 no está disponible o B0 no está utilizando modo IBC), no insertar el vector de bloque del bloque B0 en la lista de candidatos de fusión IBC

del bloque actual (recortar B0 y B1).

Si el bloque contiguo A0 está disponible y está utilizando modo IBC, recortar (o determinar) si el vector de bloque de A0 es el mismo que A1. Si el vector de bloque de A0 no es el mismo que el vector de bloque de A1, insertar el vector de bloque del bloque A0 en la lista de candidatos de fusión IBC del bloque actual. De otra manera (el vector de bloque de A0 es el mismo que el vector de bloque A1, o A0 no está disponible o A0 no está utilizando modo IBC), no insertar el vector de bloque del bloque A0 en la lista de candidatos de fusión IBC del bloque actual (recortar A0 y A1).

Si el bloque contiguo B2 está disponible y está utilizando modo IBC, y el tamaño de lista de fusión IBC actual es más pequeño que 4, recortar (o determinar) si el vector de bloque de B2 es el mismo que A1 y si el vector de bloque de B2 es el mismo que B1. Si el vector de bloque de B2 no es el mismo que el vector de bloque de A1 y el vector de bloque de B2 no es el mismo que B1, insertar el vector de bloque del bloque B2 en la lista de candidatos de fusión IBC del bloque actual. De otra manera (el vector de bloque de B2 es el mismo que el vector de bloque A1 o B1, o B0 no está disponible o B0 no está utilizando modo IBC), no insertar el vector de bloque del bloque B0 en la lista de candidatos de fusión IBC del bloque actual (recortar B2 y A1 y recortar B2 y B1).

Si el último candidato H_k de HMVP está disponible y está utilizando modo IBC, y el tamaño de lista de candidatos de fusión IBC actual es más pequeño que un valor, el valor igual al número máximo de candidatos de fusión IBC menos uno, y el vector de bloque de H_k no es el mismo que A1, insertar el vector de bloque de H_k en la lista de candidatos de fusión IBC del bloque actual. De otra manera (H_k no está disponible o H_k no está utilizando el modo IBC o el tamaño de la lista de candidatos de fusión IBC actual no es más pequeño que el número máximo de candidatos de fusión IBC menos uno, o el vector de bloque de H_k es el mismo que el vector de bloque de A1), no insertar el vector de bloque de H_k en la lista de candidatos de fusión IBC del bloque actual (recortar H_k y A1).

Si el penúltimo candidato HMVP H_{k-1} está disponible y está utilizando modo IBC, y el tamaño de lista de candidatos de fusión IBC actual es más pequeño que un valor, el valor igual al número máximo de candidatos de fusión IBC menos uno, y el vector de bloque de H_{k-1} no es el mismo que A1, insertar el vector de bloque de H_{k-1} en la lista de candidatos de fusión IBC del bloque actual. De otra manera (H_{k-1} no está disponible o H_{k-1} no está utilizando el modo IBC o el tamaño de la lista de candidatos de fusión IBC actual no es más pequeño que el número máximo de candidatos de fusión IBC menos uno, o el vector de bloque de H_{k-1} es el mismo que el vector de bloque de A1), no insertar el vector de bloque de H_{k-1} en la lista de candidatos de fusión IBC del bloque actual (recortar H_{k-1} y B1).

Insertar los candidatos HMVP restantes uno por uno, si el candidato está disponible y está utilizando modo IBC y el tamaño de lista de candidatos de fusión IBC actual es más pequeño que el número máximo de candidatos de fusión IBC menos uno (no recortar).

Insertar el candidato a modo de pares, el tamaño de lista de candidatos de fusión IBC actual es más pequeño que el número máximo de candidatos de fusión IBC (no recortar).

Solución 5

Según la solución 5, el recorte de vector de bloque del bloque contiguo espacial se mantiene igual. Para el recorte de candidato HMVP, el vector de bloque del último candidato H_k de HMVP y el penúltimo candidato H_{k-1} de HMVP es recortado con el bloque contiguo espacial del vector de bloque B1. El recorte de candidato HMVP restante se elimina.

El siguiente recorte es posible cuando se realiza durante la construcción de la lista de fusión IBC:

- Recorte entre A1 y B1
- Recorte entre A0 y A1
- Recorte entre B0 y B1
- Recorte entre B2 y A1
- Recorte entre B2 y B1
- Recorte entre el último candidato H_k de HMVP y B1
- Recorte entre el penúltimo candidato H_{k-1} de HMVP y B1.

En este caso, el número máximo de recorte para construir la lista de candidatos de fusión IBC del bloque actual se reduce de 9 a 7. La solución redujo significativamente la complejidad de la construcción de la lista de fusión IBC, tanto para el codificador, como para el decodificador.

5

En un ejemplo, la lista de candidatos de fusión IBC concreta es construida de la siguiente manera:

Si el bloque contiguo A1 está disponible y está utilizando modo IBC, insertar el vector de bloque del bloque A1 en la lista de candidatos de fusión IBC del bloque actual. De otra manera (A1 no está disponible o A1 no está utilizando modo IBC), no insertar el vector de bloque del bloque A1 en la lista de candidatos de fusión IBC del bloque actual (primer candidato, no recortar).

Si el bloque contiguo B1 está disponible y está utilizando modo IBC, recortar (o determinar) si el vector de bloque de B1 es el mismo que A1. Si el vector de bloque de B1 no es el mismo que el vector de bloque de A1, insertar el vector de bloque del bloque B1 en la lista de candidatos de fusión IBC del bloque actual. De otra manera (el vector de bloque de B1 es el mismo que el vector de bloque A1, o B1 no está disponible o B1 no está utilizando modo IBC), no insertar el vector de bloque del bloque A1 en la lista de candidatos de fusión IBC del bloque actual (recortar A1 y B1).

Si el bloque contiguo B0 está disponible y está utilizando modo IBC, recortar (o determinar) si el vector de bloque de B0 es el mismo que B1. Si el vector de bloque de B0 no es el mismo que el vector de bloque de B1, insertar el vector de bloque del bloque B0 en la lista de candidatos de fusión IBC del bloque actual. De otra manera (el vector de bloque de B0 es el mismo que el vector de bloque B1, o B0 no está disponible o B0 no está utilizando modo IBC), no insertar el vector de bloque del bloque B0 en la lista de candidatos de fusión IBC del bloque actual (recortar B0 y B1).

Si el bloque contiguo A0 está disponible y está utilizando modo IBC, recortar (o determinar) si el vector de bloque de A0 es el mismo que A1. Si el vector de bloque de A0 no es el mismo que el vector de bloque de A1, insertar el vector de bloque del bloque A0 en la lista de candidatos de fusión IBC del bloque actual. De otra manera (el vector de bloque de A0 es el mismo que el vector de bloque A1, o A0 no está disponible o A0 no está utilizando modo IBC), no insertar el vector de bloque del bloque A0 en la lista de candidatos de fusión IBC del bloque actual (recortar A0 y A1).

Si el bloque contiguo B2 está disponible y está utilizando modo IBC y el tamaño de lista de fusión IBC actual es más pequeño que 4, recortar (o determinar) si el vector de bloque de B2 es el mismo que A1 y si el vector de bloque de B2 es el mismo que B1. Si el vector de bloque de B2 no es el mismo que el vector de bloque de A1 y el vector de bloque de B2 no es el mismo que B1, insertar el vector de bloque del bloque B2 en la lista de candidatos de fusión IBC del bloque actual. De otra manera (el vector de bloque de B2 es el mismo que el vector de bloque A1 o B1, o B0 no está disponible o B0 no está utilizando modo IBC), no insertar el vector de bloque del bloque B0 en la lista de candidatos de fusión IBC del bloque actual (recortar B2 y A1 y recortar B2 y B1).

Si el último candidato H_k de HMVP está disponible y está utilizando modo IBC y el tamaño de lista de candidatos de fusión IBC actual es más pequeño que un valor, el valor igual al número máximo de candidatos de fusión IBC menos uno, y el vector de bloque de H_k no es el mismo que B1, insertar el vector de bloque de H_k en la lista de candidatos de fusión IBC del bloque actual. De otra manera (H_k no está disponible o H_k no está utilizando el modo IBC o el tamaño de la lista de candidatos de fusión IBC actual no es más pequeño que el número máximo de candidatos de fusión IBC menos uno, o el vector de bloque de H_k es el mismo que el vector de bloque de B1), no insertar el vector de bloque de H_k en la lista de candidatos de fusión IBC del bloque actual (recortar H_k y B1).

Si el penúltimo candidato HMVP H_{k-1} está disponible y está utilizando modo IBC, y el tamaño de lista de candidatos de fusión IBC actual es más pequeño que un valor, el valor igual al número máximo de candidatos de fusión IBC menos uno, y el vector de bloque de H_{k-1} no es el mismo que B1, insertar el vector de bloque de H_{k-1} en la lista de candidatos de fusión IBC del bloque actual. De otra manera (H_{k-1} no está disponible o H_{k-1} no está utilizando el modo IBC o el tamaño de la lista de candidatos de fusión IBC actual no es más pequeño que el número máximo de candidatos de fusión IBC menos uno, o el vector de bloque de H_{k-1} es el mismo que el vector de bloque de B1), no insertar el vector de bloque de H_{k-1} en la lista de candidatos de fusión IBC del bloque actual (recortar H_{k-1} y B1).

Insertar los candidatos HMVP restantes uno por uno, si el candidato está disponible y está utilizando modo IBC y el tamaño de lista de candidatos de fusión IBC actual es más pequeño que el número máximo de candidatos de fusión IBC menos uno (no recortar).

Insertar el candidato a modo de pares, el tamaño de lista de candidatos de fusión IBC actual es más pequeño que el número máximo de candidatos de fusión IBC máximo (no recortar).

Solución 6

- Según la solución 6, en esta solución el último candidato de la lista de candidatos de fusión IBC es permitido utilizando el modo HMVP, el proceso de recorte de la construcción de lista de fusión IBC en esta solución no se modifica, pero este método introdujo un método de construcción de lista de fusión más eficiente.

En un ejemplo, la lista concreta de candidatos de fusión IBC es construida de la siguiente manera:

- 10 Si el bloque contiguo A1 está disponible y está utilizando modo IBC, insertar el vector de bloque del bloque A1 en la lista de candidatos de fusión IBC del bloque actual. De otra manera (A1 no está disponible o A1 no está utilizando modo IBC), no insertar el vector de bloque del bloque A1 en la lista de candidatos de fusión IBC del bloque actual (primer candidato, no recortar).
- 15 Si el bloque contiguo B1 está disponible y está utilizando modo IBC, recortar (o determinar) si el vector de bloque de B1 es el mismo que A1. Si el vector de bloque de B1 no es el mismo que el vector de bloque de A1, insertar el vector de bloque del bloque B1 en la lista de candidatos de fusión IBC del bloque actual. De otra manera (el vector de bloque de B1 es el mismo que el vector de bloque A1, o B1 no está disponible o B1 no está utilizando modo IBC), no insertar el vector de bloque del bloque A1 en la lista de candidatos de fusión IBC del bloque actual (recortar A1 y B1).
- 20 Si el bloque contiguo B0 está disponible y está utilizando modo IBC, recortar (o determinar) si el vector de bloque de B0 es el mismo que B1. Si el vector de bloque de B0 no es el mismo que el vector de bloque de B1, insertar el vector de bloque del bloque B0 en la lista de candidatos de fusión IBC del bloque actual. De otra manera (el vector de bloque de B0 es el mismo que el vector de bloque B1, o B0 no está disponible o B0 no está utilizando modo IBC), no insertar el vector de bloque del bloque B0 en la lista de candidatos de fusión IBC del bloque actual (recortar B0 y B1).
- 25 Si el bloque contiguo A0 está disponible y está utilizando modo IBC, recortar (o determinar) si el vector de bloque de A0 es el mismo que A1. Si el vector de bloque de A0 no es el mismo que el vector de bloque de A1, insertar el vector de bloque del bloque A0 en la lista de candidatos de fusión IBC del bloque actual. De otra manera (el vector de bloque de A0 es el mismo que el vector de bloque A1, o A0 no está disponible o A0 no está utilizando modo IBC), no insertar el vector de bloque del bloque A0 en la lista de candidatos de fusión IBC del bloque actual (recortar A0 y A1).
- 30 Si el bloque contiguo B2 está disponible y está utilizando modo IBC, y el tamaño de lista de fusión IBC actual es más pequeño que 4, recortar (o determinar) si el vector de bloque de B2 es el mismo que A1 y si el vector de bloque de B2 es el mismo que B1. Si el vector de bloque de B2 no es el mismo que el vector de bloque de A1 y el vector de bloque de B2 no es el mismo que B1, insertar el vector de bloque del bloque B2 en la lista de candidatos de fusión IBC del bloque actual. De otra manera (el vector de bloque de B2 es el mismo que el vector de bloque A1 o B1, o B0 no está disponible o B0 no está utilizando modo IBC), no insertar el vector de bloque del bloque B0 en la lista de candidatos de fusión IBC del bloque actual (recortar B2 y A1, recortar B2 y B1).
- 35 Si el último candidato H_k de HMVP está disponible y está utilizando modo IBC, y el tamaño de lista de candidatos de fusión IBC actual es más pequeño que el número máximo de candidatos de fusión IBC, y el vector de bloque de H_k no es el mismo que A1 y B1, insertar el vector de bloque de H_k en la lista de candidatos de fusión IBC del bloque actual. De otra manera (H_k no está disponible o H_k no está utilizando el modo IBC o el tamaño de la lista de candidatos de fusión IBC actual no es más pequeño que el número máximo de candidatos de fusión IBC, o el vector de bloque de H_k es el mismo que el vector de bloque de A1, o el vector de bloque de H_k es el mismo que el vector de bloque de B1), no insertar el vector de bloque de H_k en la lista de candidatos de fusión IBC del bloque actual (recortar H_k y A1, H_k y B1).
- 40 Si el penúltimo candidato H_{k-1} de HMVP está disponible y está utilizando modo IBC, y el tamaño de lista de candidatos de fusión IBC actual es más pequeño que el número máximo de candidatos de fusión IBC y el vector de bloque de H_{k-1} no es el mismo que A1 y B1, insertar el vector de bloque de H_{k-1} en la lista de candidatos de fusión IBC del bloque actual. De otra manera (H_{k-1} no está disponible o H_{k-1} no está utilizando el modo IBC o el tamaño de la lista de candidatos de fusión IBC actual no es más pequeño que el número máximo de candidatos de fusión IBC o el vector de bloque de H_{k-1} es el mismo que el vector de bloque de A1, o el vector de bloque de H_{k-1} es el mismo que el vector de bloque de B1), no insertar el vector de bloque de H_{k-1} en la lista de candidatos de fusión IBC del bloque actual (recortar H_{k-1} y A1, H_{k-1} y B1).
- 45 Insertar los candidatos HMVP restantes uno por uno, si el candidato está disponible y está utilizando modo IBC, y el tamaño de lista de candidatos de fusión IBC actual es más pequeño que el número máximo de candidatos de fusión IBC (no recortar).
- 50
- 55
- 60
- 65

Insertar el candidato a modo de pares, el tamaño de lista de candidatos de fusión IBC actual es más pequeño que el número máximo de candidatos de fusión IBC (no recortar).

Solución 7

5

Según la solución 7, el recorte de vector de bloque del bloque contiguo espacial se mantiene igual. Para el recorte de candidato HMVP, el vector de bloque del último candidato H_k y de HMVP el penúltimo candidato H_{k-1} de HMVP es recortado con el vector de bloque del primer bloque contiguo espacial de A1 y B1. Si ninguno de A1 o B1 ya está dentro de la lista de fusión IBC, entonces ningún candidato HMVP es probado con el candidato espacial.

10

El recorte de candidato HMVP restante se elimina. El siguiente recorte es el proceso de recorte del peor caso posible durante la construcción de la lista de fusión IBC:

15

- Recorte entre A1 y B1

- Recorte entre A0 y A1

20

- Recorte entre B0 y B1

- Recorte entre B2 y A1

- Recorte entre B2 y B1

25

- Recorte entre el último candidato H_k de HMVP y A1 o H_k y B1

- Recorte entre el penúltimo candidato H_{k-1} de HMVP y A1 o H_{k-1} y B1.

30

En este caso, el número máximo de recorte para construir la lista de candidatos de fusión IBC del bloque actual se reduce de 9 a 7. La solución redujo significativamente la complejidad de la construcción de la lista de fusión IBC para el codificador y el decodificador.

En un ejemplo, la lista de candidatos de fusión IBC concreta es construida de la siguiente manera:

35

Si el bloque contiguo A1 está disponible y está utilizando modo IBC, insertar el vector de bloque del bloque A1 en la lista de candidatos de fusión IBC del bloque actual. De otra manera (A1 no está disponible o A1 no está utilizando modo IBC), no insertar el vector de bloque del bloque A1 en la lista de candidatos de fusión IBC del bloque actual (primer candidato, no recortar).

40

Si el bloque contiguo B1 está disponible y está utilizando modo IBC, recortar (o determinar) si el vector de bloque de B1 es el mismo que A1. Si el vector de bloque de B1 no es el mismo que el vector de bloque de A1, insertar el vector de bloque del bloque B1 en la lista de candidatos de fusión IBC del bloque actual. De otra manera (el vector de bloque de B1 es el mismo que el vector de bloque A1, o B1 no está disponible o B1 no está utilizando modo IBC), no insertar el vector de bloque del bloque A1 en la lista de candidatos de fusión IBC del bloque actual (recortar A1 y B1).

45

Si el bloque contiguo B0 está disponible y está utilizando modo IBC, recortar (o determinar) si el vector de bloque de B0 es el mismo que B1. Si el vector de bloque de B0 no es el mismo que el vector de bloque de B1, insertar el vector de bloque del bloque B0 en la lista de candidatos de fusión IBC del bloque actual. De otra manera (el vector de bloque de B0 es el mismo que el vector de bloque B1, o B0 no está disponible o B0 no está utilizando modo IBC), no insertar el vector de bloque del bloque B0 en la lista de candidatos de fusión IBC del bloque actual (recortar B0 y B1).

50

Si el bloque contiguo A0 está disponible y está utilizando modo IBC, recortar (o determinar) si el vector de bloque de A0 es el mismo que A1. Si el vector de bloque de A0 no es el mismo que el vector de bloque de A1, insertar el vector de bloque del bloque A0 en la lista de candidatos de fusión IBC del bloque actual. De otra manera (el vector de bloque de A0 es el mismo que el vector de bloque A1, o A0 no está disponible o A0 no está utilizando modo IBC), no insertar el vector de bloque del bloque A0 en la lista de candidatos de fusión IBC del bloque actual (recortar A0 y A1).

55

60

Si el bloque contiguo B2 está disponible y está utilizando modo IBC y el tamaño de lista de fusión IBC actual es más pequeño que 4, recortar (o determinar) si el vector de bloque de B2 es el mismo que A1 y si el vector de bloque de B2 es el mismo que B1. Si el vector de bloque de B2 no es el mismo que el vector de bloque de A1 y el vector de bloque de B2 no es el mismo que B1, insertar el vector de bloque del bloque B2 en la lista de candidatos de fusión IBC del bloque actual. De otra manera (el vector de bloque de B2 es el mismo que el

65

vector de bloque A1 o B1, o B0 no está disponible o B0 no está utilizando modo IBC), no insertar el vector de bloque del bloque B0 en la lista de candidatos de fusión IBC del bloque actual (recortar B2 y A1 y recortar B2 y B1).

5 Si A1 ya está en la lista de fusión IBC, entonces

Si el último candidato H_k de HMVP está disponible y está utilizando modo IBC y el tamaño de lista de candidatos de fusión IBC actual es más pequeño que un valor, el valor igual al número máximo de candidatos de fusión IBC menos uno, y el vector de bloque de H_k no es el mismo que A1, insertar el vector de bloque de H_k en la lista de candidatos de fusión IBC del bloque actual. De otra manera (H_k no está disponible o H_k no está utilizando el modo IBC o el tamaño de la lista de candidatos de fusión IBC actual no es más pequeño que el número máximo de candidatos de fusión IBC menos uno, o el vector de bloque de H_k es el mismo que el vector de bloque de A1), no insertar el vector de bloque de H_k en la lista de candidatos de fusión IBC del bloque actual (recortar H_k y A1).

Si el penúltimo candidato HMVP H_{k-1} está disponible y está utilizando modo IBC, y el tamaño de lista de candidatos de fusión IBC actual es más pequeño que un valor, el valor igual al número máximo de candidatos de fusión IBC menos uno, y el vector de bloque de H_{k-1} no es el mismo que A1, insertar el vector de bloque de H_{k-1} en la lista de candidatos de fusión IBC del bloque actual. De otra manera (H_{k-1} no está disponible o H_{k-1} no está utilizando el modo IBC o el tamaño de la lista de candidatos de fusión IBC actual no es más pequeño que el número máximo de candidatos de fusión IBC menos uno, o el vector de bloque de H_{k-1} es el mismo que el vector de bloque de A1), no insertar el vector de bloque de H_{k-1} en la lista de candidatos de fusión IBC del bloque actual (recortar H_{k-1} y A1).

25 De otra manera si A1 no está en la lista de fusión IBC y B1 ya está en la lista de fusión IBC, entonces

Si el último candidato H_k de HMVP está disponible y está utilizando modo IBC y el tamaño de lista de candidatos de fusión IBC actual es más pequeño que un valor, el valor igual al número máximo de candidatos de fusión IBC menos uno, y el vector de bloque de H_k no es el mismo que B1, insertar el vector de bloque de H_k en la lista de candidatos de fusión IBC del bloque actual. De otra manera (H_k no está disponible o H_k no está utilizando el modo IBC o el tamaño de la lista de candidatos de fusión IBC actual no es más pequeño que el número máximo de candidatos de fusión IBC menos uno, o el vector de bloque de H_k es el mismo que el vector de bloque de B1), no insertar el vector de bloque de H_k en la lista de candidatos de fusión IBC del bloque actual (recortar H_k y B1).

Si el penúltimo candidato HMVP H_{k-1} está disponible y está utilizando modo IBC, y el tamaño de lista de candidatos de fusión IBC actual es más pequeño que un valor, el valor igual al número máximo de candidatos de fusión IBC menos uno, y el vector de bloque de H_{k-1} no es el mismo que B1, insertar el vector de bloque de H_{k-1} en la lista de candidatos de fusión IBC del bloque actual. De otra manera (H_{k-1} no está disponible o H_{k-1} no está utilizando el modo IBC o el tamaño de la lista de candidatos de fusión IBC actual no es más pequeño que el número máximo de candidatos de fusión IBC menos uno, o el vector de bloque de H_{k-1} es el mismo que el vector de bloque de B1), no insertar el vector de bloque de H_{k-1} en la lista de candidatos de fusión IBC del bloque actual (recortar H_{k-1} y B1).

45 De otra manera si ninguno de A1 y B1 está listo en la lista de fusión IBC entonces,

Si el último candidato HMVP H_k está disponible y está utilizando modo IBC y el tamaño de lista de candidatos de fusión IBC actual es más pequeño que un valor, el valor igual al número máximo de candidatos de fusión IBC menos uno, insertar el vector de bloque de H_k en la lista de candidatos de fusión IBC del bloque actual. De otra manera (H_k no está disponible o H_k no está utilizando el modo IBC o el tamaño de la lista de candidatos de fusión IBC actual no es más pequeño que el número máximo de candidatos de fusión IBC menos uno), no insertar el vector de bloque de H_k en la lista de candidatos de fusión IBC del bloque actual (no recortar).

Si el penúltimo candidato HMVP H_{k-1} está disponible y está utilizando modo IBC, y el tamaño de lista de candidatos de fusión IBC actual es más pequeño que un valor, el valor igual al número máximo de candidatos de fusión IBC menos uno, insertar el vector de bloque de H_{k-1} en la lista de candidatos de fusión IBC del bloque actual. De otra manera (H_{k-1} no está disponible o H_{k-1} no está utilizando el modo IBC o el tamaño de la lista de candidatos de fusión IBC actual no es más pequeño que el número máximo de candidatos de fusión IBC menos uno), no insertar el vector de bloque de H_{k-1} en la lista de candidatos de fusión IBC del bloque actual (no recortar).

Insertar los candidatos HMVP restantes uno por uno, si el candidato está disponible y está utilizando modo IBC y el tamaño de lista de candidatos de fusión IBC actual es más pequeño que el número máximo de candidatos de fusión IBC menos uno (no recortar).

Insertar el candidato a modo de pares, el tamaño de lista de candidatos de fusión IBC actual es más pequeño

que el número máximo de candidatos de fusión IBC (no recortar).

Solución 8

- 5 Según la solución 8, el recorte de vector de bloque del bloque contiguo espacial se mantiene igual. Para el recorte de candidato HMVP,

10 el vector de bloque del último candidato H_k de HMVP y el penúltimo candidato H_{k-1} de HMVP son recortados con el vector de bloque del bloque contiguo espacial de A1, cuando A1 ya está en la lista de fusión IBC, y B1 no está en la lista de fusión IBC;

15 el vector de bloque del último candidato H_k de HMVP y el penúltimo candidato H_{k-1} de HMVP son recortados con el vector de bloque del bloque contiguo espacial de B1, cuando B1 ya está en la lista de fusión IBC, y A1 no está en la lista de fusión IBC;

20 el vector de bloque del último candidato H_k de HMVP es recortado con el vector de bloque del bloque contiguo espacial de B1, cuando ambos A1 y B1 ya están en la lista de fusión IBC;

Si ninguno de A1 o B1 ya está dentro de la lista de fusión IBC, entonces ningún candidato HMVP es probado con el candidato espacial.

25 En este caso, el número máximo de recorte para construir la lista de candidatos de fusión IBC del bloque actual se reduce de 9 a 7. La solución redujo significativamente la complejidad de la construcción de la lista de fusión IBC para el codificador y el decodificador.

En un ejemplo, la lista concreta de candidatos de fusión IBC es construida de la siguiente manera:

30 Si el bloque contiguo A1 está disponible y está utilizando modo IBC, insertar el vector de bloque del bloque A1 en la lista de candidatos de fusión IBC del bloque actual. De otra manera (A1 no está disponible o A1 no está utilizando modo IBC), no insertar el vector de bloque del bloque A1 en la lista de candidatos de fusión IBC del bloque actual (primer candidato, no recortar).

35 Si el bloque contiguo B1 está disponible y está utilizando modo IBC, recortar (o determinar) si el vector de bloque de B1 es el mismo que A1. Si el vector de bloque de B1 no es el mismo que el vector de bloque de A1, insertar el vector de bloque del bloque B1 en la lista de candidatos de fusión IBC del bloque actual. De otra manera (el vector de bloque de B1 es el mismo que el vector de bloque de A1, o B1 no está disponible o B1 no está utilizando modo IBC), no insertar el vector de bloque del bloque A1 en la lista de candidatos de fusión IBC del bloque actual (recortar A1 y B1).

40 Si el bloque contiguo B0 está disponible y está utilizando modo IBC, recortar (o determinar) si el vector de bloque de B0 es el mismo que B1. Si el vector de bloque de B0 no es el mismo que el vector de bloque de B1, insertar el vector de bloque del bloque B0 en la lista de candidatos de fusión IBC del bloque actual. De otra manera (el vector de bloque de B0 es el mismo que el vector de bloque de B1, o B0 no está disponible o B0 no está utilizando modo IBC), no insertar el vector de bloque del bloque B0 en la lista de candidatos de fusión IBC del bloque actual (recortar B0 y B1).

50 Si el bloque contiguo A0 está disponible y está utilizando modo IBC, recortar (o determinar) si el vector de bloque de A0 es el mismo que A1. Si el vector de bloque de A0 no es el mismo que el vector de bloque de A1, insertar el vector de bloque del bloque A0 en la lista de candidatos de fusión IBC del bloque actual. De otra manera (el vector de bloque de A0 es el mismo que el vector de bloque de A1, o A0 no está disponible o A0 no está utilizando modo IBC), no insertar el vector de bloque del bloque A0 en la lista de candidatos de fusión IBC del bloque actual (recortar A0 y A1).

55 Si el bloque contiguo B2 está disponible y está utilizando modo IBC y el tamaño de lista de fusión IBC actual es más pequeño que 4, recortar (o determinar) si el vector de bloque de B2 es el mismo que A1 y si el vector de bloque de B2 es el mismo que B1. Si el vector de bloque de B2 no es el mismo que el vector de bloque de A1 y el vector de bloque de B2 no es el mismo que B1, insertar el vector de bloque del bloque B2 en la lista de candidatos de fusión IBC del bloque actual. De otra manera (el vector de bloque de B2 es el mismo que el vector de bloque de A1 o B1, o B0 no está disponible o B0 no está utilizando modo IBC), no insertar el vector de bloque del bloque B0 en la lista de candidatos de fusión IBC del bloque actual (recortar B2 y A1 y recortar B2 y B1).

Si A1 ya está en la lista de fusión IBC y B1 no está en la lista, entonces

65 Si el último candidato HMVP H_k está disponible y está utilizando modo IBC y el tamaño de lista de candidatos de fusión IBC actual es más pequeño que un valor, el valor igual al número máximo de candidatos de fusión

IBC menos uno, y el vector de bloque de H_k no es el mismo que A_1 , insertar el vector de bloque de H_k en la lista de candidatos de fusión IBC del bloque actual. De otra manera (H_k no está disponible o H_k no está utilizando el modo IBC o el tamaño de la lista de candidatos de fusión IBC actual no es más pequeño que el número máximo de candidatos de fusión IBC menos uno, o el vector de bloque de H_k es el mismo que el vector de bloque de A_1), no insertar el vector de bloque de H_k en la lista de candidatos de fusión IBC del bloque actual (recortar H_k y A_1).

Si el penúltimo candidato HMVP H_{k-1} está disponible y está utilizando modo IBC, y el tamaño de lista de candidatos de fusión IBC actual es más pequeño que un valor, el valor igual al número máximo de candidatos de fusión IBC menos uno, y el vector de bloque de H_{k-1} no es el mismo que A_1 , insertar el vector de bloque de H_{k-1} en la lista de candidatos de fusión IBC del bloque actual. De otra manera (H_{k-1} no está disponible o H_{k-1} no está utilizando el modo IBC o el tamaño de la lista de candidatos de fusión IBC actual no es más pequeño que el número máximo de candidatos de fusión IBC menos uno, o el vector de bloque de H_{k-1} es el mismo que el vector de bloque de A_1), no insertar el vector de bloque de H_{k-1} en la lista de candidatos de fusión IBC del bloque actual (recortar H_{k-1} y A_1).

De otra manera si A_1 no está en la lista de fusión IBC y B_1 ya está en la lista de fusión IBC, entonces

Si el último candidato H_k de HMVP está disponible y está utilizando modo IBC y el tamaño de lista de candidatos de fusión IBC actual es más pequeño que un valor, el valor igual al número de candidatos de fusión IBC máximo menos uno, y el vector de bloque de H_k no es el mismo que B_1 , insertar el vector de bloque de H_k en la lista de candidatos de fusión IBC del bloque actual. De otra manera (H_k no está disponible o H_k no está utilizando el modo IBC o el tamaño de la lista de candidatos de fusión IBC actual no es más pequeño que el número máximo de candidatos de fusión IBC menos uno, o el vector de bloque de H_k es el mismo que el vector de bloque de B_1), no insertar el vector de bloque de H_k en la lista de candidatos de fusión IBC del bloque actual (recortar H_k y B_1).

Si el penúltimo candidato H_{k-1} de HMVP está disponible y está utilizando modo IBC, y el tamaño de lista de candidatos de fusión IBC actual es más pequeño que un valor, el valor igual al número máximo de candidatos de fusión IBC menos uno, y el vector de bloque de H_{k-1} no es el mismo que B_1 , insertar el vector de bloque de H_{k-1} en la lista de candidatos de fusión IBC del bloque actual. De otra manera (H_{k-1} no está disponible o H_{k-1} no está utilizando el modo IBC o el tamaño de la lista de candidatos de fusión IBC actual no es más pequeño que el número máximo de candidatos de fusión IBC menos uno, o el vector de bloque de H_{k-1} es el mismo que el vector de bloque de B_1), no insertar el vector de bloque de H_{k-1} en la lista de candidatos de fusión IBC del bloque actual (recortar H_{k-1} y B_1).

De otra manera si ambos A_1 y B_1 ya están en la lista de fusión IBC, entonces

Si el último candidato H_k de HMVP está disponible y está utilizando modo IBC y el tamaño de lista de candidatos de fusión IBC actual es más pequeño que un valor, el valor igual al número máximo de candidatos de fusión IBC menos uno, y el vector de bloque de H_k no es el mismo que A_1 y el vector de bloque de H_k no es el mismo que B_1 , insertar el vector de bloque de H_k en la lista de candidatos de fusión IBC del bloque actual. De otra manera (H_k no está disponible o H_k no está utilizando el modo IBC o el tamaño de la lista de candidatos de fusión IBC actual no es más pequeño que el número máximo de candidatos de fusión IBC menos uno, o el vector de bloque de H_k es el mismo que el vector de bloque de A_1 o el vector de bloque de H_k es el mismo que el vector de bloque de B_1), no insertar el vector de bloque de H_k en la lista de candidatos de fusión IBC del bloque actual (recortar H_k y A_1 y B_1).

De otra manera si ninguno de A_1 y B_1 está listo en la lista de fusión IBC entonces,

Si el último candidato H_k de HMVP está disponible y está utilizando modo IBC y el tamaño de lista de candidatos de fusión IBC actual es más pequeño que un valor, el valor igual al número máximo de candidatos de fusión IBC menos uno, insertar el vector de bloque de H_k en la lista de candidatos de fusión IBC del bloque actual. De otra manera (H_k no está disponible o H_k no está utilizando el modo IBC o el tamaño de la lista de candidatos de fusión IBC actual no es más pequeño que el número máximo de candidatos de fusión IBC menos uno), no insertar el vector de bloque de H_k en la lista de candidatos de fusión IBC del bloque actual (no recortar).

Si el penúltimo candidato HMVP H_{k-1} está disponible y está utilizando modo IBC, y el tamaño de lista de candidatos de fusión IBC actual es más pequeño que un valor, el valor igual al número máximo de candidatos de fusión IBC menos uno, insertar el vector de bloque de H_{k-1} en la lista de candidatos de fusión IBC del bloque actual. De otra manera (H_{k-1} no está disponible o H_{k-1} no está utilizando el modo IBC o el tamaño de la lista de candidatos de fusión IBC actual no es más pequeño que el número máximo de candidatos de fusión IBC menos uno), no insertar el vector de bloque de H_{k-1} en la lista de candidatos de fusión IBC del bloque actual (no recortar).

Insertar los candidatos HMVP restantes uno por uno, si el candidato está disponible y está utilizando modo IBC

y el tamaño de lista de candidatos de fusión IBC actual es más pequeño que el número máximo de candidatos de fusión IBC menos uno (no recortar).

- 5 Insertar el candidato en pares, el tamaño de lista de candidatos de fusión IBC actual es más pequeño que el número máximo de candidatos de fusión IBC (no recortar).

Solución 9

- 10 Según la solución 9, encima de la solución 1 el contiguo espacial B0 basado en la disponibilidad de B0 y B1; el contiguo espacial A0 basado en la disponibilidad de A0 y A1; el contiguo espacial B2 basado en la disponibilidad de B2, B1 y A1,

En el ejemplo,

- 15 Si el bloque contiguo B0 está disponible y está utilizando modo IBC y B1 no está disponible, el vector de bloque del bloque B0 es insertado en la lista de candidatos de fusión IBC del bloque actual. De otra manera (B0 no está disponible o B0 no está utilizando el modo IBC o B1 está disponible), el vector de bloque del bloque B0 no es insertado en la lista de candidatos de fusión IBC del bloque actual (no recortar).
- 20 Si el bloque contiguo A0 está disponible y está utilizando modo IBC y A1 no está disponible, el vector de bloque del bloque A0 es insertado en la lista de candidatos de fusión IBC del bloque actual. De otra manera (A0 no está disponible o A0 no está utilizando el modo IBC o A1 está disponible), el vector de bloque del bloque A0 no es insertado en la lista de candidatos de fusión IBC del bloque actual (no recortar).
- 25 Si el bloque contiguo B2 está disponible y está utilizando modo IBC, ambos A1 y B1 no están disponibles, y el tamaño de la lista de candidatos de fusión IBC actual es más pequeño que 4, el vector de bloque del bloque B2 es insertado en la lista de candidatos de fusión IBC del bloque actual. De otra manera (B2 no está disponible o B2 no está utilizando el modo IBC o el tamaño de la lista de candidatos de fusión IBC actual no es más pequeño que 4 o A1 está disponible o B1 está disponible), el vector de bloque del bloque B2 no es insertado
- 30 en la lista de candidatos de fusión IBC del bloque actual (no recortar).

En esta solución, no se añade ningún recorte adicional, pero se logra una mayor eficiencia de la codificación en comparación con la solución 1.

- 35 Solución 10

Según la solución 10, cualquiera de las soluciones 1 a 9 se puede combinar para reducir la complejidad de la construcción de la lista de fusión IBC.

- 40 En un ejemplo para la solución 10, se combinan la solución 2 y la solución 3. Debido a que los bloques contiguos espaciales a la izquierda (A1) y por encima (B1) son importantes para predecir el bloque actual utilizando el modo IBC. En este ejemplo, el recorte del bloque contiguo espacial entre A1 y B1 se mantiene, los vectores de bloque del bloque contiguo espacial restante no son insertados en la lista de fusión IBC. Para el recorte de candidato HMVP, el vector de bloque del último candidato H_k de HMVP es recortado con el bloque contiguo
- 45 espacial del vector de bloque A1 y B1. En una modalidad, el siguiente recorte es posible cuando se realiza durante la construcción de la lista de fusión IBC:

- Recorte entre A1 y B1
- 50 • Recorte entre el último candidato H_k de HMVP y A1
- Recorte entre el último candidato H_k de HMVP y B1.

- 55 En este caso, el número máximo de recortes para construir la lista de candidatos de fusión IBC del bloque actual se reduce de 9 a 3. La solución redujo significativamente la complejidad de la construcción de la lista de fusión IBC para el codificador y el decodificador.

En un ejemplo, el último candidato de la lista de candidatos de fusión IBC es permitido utilizando el modo HMVP.

- 60 En un ejemplo, la lista concreta de candidatos de fusión IBC es construida de la siguiente manera:

- 65 Si el bloque contiguo A1 está disponible y está utilizando modo IBC, el vector de bloque del bloque A1 es insertado en la lista de candidatos de fusión IBC del bloque actual. De otra manera (A1 no está disponible o A1 no está utilizando modo IBC), el vector de bloque del bloque A1 no es insertado en la lista de candidatos de fusión IBC del bloque actual (primer candidato, no recortar).

- Si el bloque contiguo B1 está disponible y está utilizando modo IBC y el vector de bloque del bloque A1 es insertado en la lista de candidatos de fusión IBC del bloque actual, recortar (o determinar) si el vector de bloque de B1 es el mismo que A1. Si el vector de bloque de B1 no es el mismo que el vector de bloque de A1, el vector de bloque del bloque B1 es insertado en la lista de candidatos de fusión IBC del bloque actual. De otra manera (el vector de bloque de B1 es el mismo que el vector de bloque A1, o B1 no está disponible o B1 no está utilizando modo IBC), el vector de bloque del bloque B1 no es insertado en la lista de candidatos de fusión IBC del bloque actual (recortar A1 y B1).
- Si el último candidato H_k de HMVP está disponible y está utilizando modo IBC, y el tamaño de lista de candidatos de fusión IBC actual es más pequeño que un valor, el valor igual al número máximo de candidatos de fusión IBC menos uno; y el vector de bloque de H_k no es el mismo que A1 y B1, insertar el vector de bloque de H_k en la lista de candidatos de fusión IBC del bloque actual. De otra manera (H_k no está disponible o H_k no está utilizando el modo IBC o el tamaño de la lista de candidatos de fusión IBC actual no es más pequeño que el número máximo de candidatos de fusión IBC menos uno, o el vector de bloque de H_k es el mismo que el vector de bloque de A1, o el vector de bloque de H_k es el mismo que el vector de bloque de B1) no insertar el vector de bloque de H_k en la lista de candidatos de fusión IBC del bloque actual (recortar H_k y A1, H_k y B1).
- Insertar los candidatos HMVP restantes uno por uno, si el candidato está disponible y está utilizando modo IBC, y el tamaño de lista de candidatos de fusión IBC actual es más pequeño que el número máximo de candidatos de fusión IBC menos uno (no recortar).
- En algunos ejemplos, además insertar el candidato a modo de pares, el tamaño de lista de candidatos de fusión IBC actual es más pequeño que el número máximo de candidatos de fusión IBC (no recortar).
- En un ejemplo, para el proceso de decodificación para unidades de codificación codificadas en el modo de predicción IBC.
- ### 8.6.1 Proceso de decodificación general para unidades de codificación codificadas en el modo de predicción IBC
- Las entradas a este proceso son:
- una ubicación de luma (x_{Cb} , y_{Cb}) que especifica la muestra superior-izquierda del bloque de codificación actual con relación a la muestra de luma superior izquierda de la imagen actual,
 - una variable $cb\ Width$ que especifica la anchura del bloque de codificación actual en muestras de luma,
 - una variable $cb\ Height$ que especifica la altura del bloque de codificación actual en muestras de luma,
 - una variable $treeType$ que especifica si se utiliza un árbol único o un árbol doble y en caso de utilizarse un árbol doble, especifica si el árbol actual corresponde a los componentes de luma o croma.
- El resultado de este proceso es una imagen reconstruida modificada antes de la filtración en-bucle.
- El proceso de derivación para parámetros de cuantificación es consultado con la ubicación de luma (x_{Cb} , y_{Cb}), la anchura del bloque de codificación actual en muestras de luma $cbWidth$ y la altura del bloque de codificación actual en muestras de luma $cbHeight$, y la variable $treeType$ como entradas.
- La variable $IsGt4by4$ es derivada de la siguiente manera:
- $$IsGt4by4 = (cbWidth * cbHeight) > 16 \quad (1111).$$
- El proceso de decodificación para unidades de codificación codificadas en modo de predicción IBC consiste de las siguientes etapas ordenadas:
1. Los componentes de vector de bloque de la unidad de codificación actual son derivados de la siguiente manera:
- El proceso de derivación para componentes de vector de bloque es consultado con la ubicación del bloque de codificación de luma (x_{Cb} , y_{Cb}), la anchura del bloque de codificación de luma $cbWidth$ y la altura del bloque de codificación de luma $cbHeight$ como entradas, y el vector de bloque de luminancia bvL como salida.
- Cuando $treeType$ es igual a ÁRBOL_ÚNICO, el proceso de derivación para el bloque de croma es consultado con el vector bvL de bloque de luminancia como entrada, y el vector bvC de bloque de croma como salida.

2. Las muestras de predicción de la unidad de codificación actual son derivadas de la siguiente manera:

El proceso de decodificación para bloques IBC conforme a lo especificado en la cláusula 8.6.3.1 es consultado con la ubicación del bloque de codificación de luma (x_{Cb} , y_{Cb}), la anchura del bloque de codificación de luma $cbWidth$ y la altura del bloque de codificación de luma $cbHeight$, el vector bvL de bloque de luma, la variable $cldx$ establecida igual a 0 como entradas, y las muestras de predicción IBC ($predSamples$) que son una agrupación $predSamplesL(cbWidth) \times (cbHeight)$ de muestras de luma de predicción como salidas.

Cuando $treeType$ es igual a ÁRBOL_ÚNICO, las muestras de predicción de la unidad de codificación actual son derivadas de la siguiente manera:

- El proceso de decodificación para bloques IBC conforme a lo especificado en la cláusula 8.6.3.1 es consultado con la ubicación del bloque de codificación de luma (x_{Cb} , y_{Cb}), la anchura del bloque de codificación de luma $cbWidth$ y la altura del bloque de codificación de luma $cbHeight$, el vector bvC de bloque de croma y la variable $cldx$ establecida igual a 1 como entradas, y las muestras de predicción IBC ($predSamples$) que son una agrupación $predSamplesCb(cbWidth / SubWidthC) \times (cbHeight / SubHeightC)$ de muestras de croma de predicción para los componentes Cb de croma como salidas.

- El proceso de decodificación para bloques IBC conforme a lo especificado en la cláusula 8.6.3.1 es consultado con la ubicación del bloque de codificación de luma (x_{Cb} , y_{Cb}), la anchura del bloque de codificación de luminancia $cbWidth$ y la altura del bloque de codificación de luma $cbHeight$, el vector bvC de bloque de croma y la variable $cldx$ establecida igual a 2 como entradas, y las muestras de predicción IBC ($predSamples$) que son una agrupación $predSamplesCr(cbWidth / SubWidthC) \times (cbHeight / SubHeightC)$ de muestras de croma de predicción para los componentes Cr de croma como salidas.

3. Las muestras residuales de la unidad de codificación actual son derivadas de la siguiente manera:

- El proceso de decodificación para la señal residual de bloques de codificación codificados en modo de inter-predicción es consultado con la ubicación (x_{Tb0} , y_{Tb0}) establecida igual a la ubicación de luma (x_{Cb} , y_{Cb}), la anchura $nTbW$ establecida igual a la anchura $cbWidth$ del bloque de codificación de luma, la altura $nTbH$ establecida igual a la altura $cbHeight$ del bloque de codificación de luma y la variable $cldx$ establecida igual a 0 como entradas, y la agrupación $resSamplesL$ como salida.

- Cuando $treeType$ es igual a ÁRBOL_ÚNICO, el proceso de decodificación para la señal residual de bloques de codificación codificados en modo de inter-predicción es consultado con la ubicación (x_{Tb0} , y_{Tb0}) establecida igual a la ubicación de croma ($x_{Cb} / SubWidthC$, $y_{Cb} / SubHeightC$), la anchura $nTbW$ establecida igual a la anchura del bloque de codificación de croma $cbWidth / SubWidthC$, la altura $nTbH$ establecida igual a la altura del bloque de codificación de croma $cbHeight / SubHeightC$ y la variable $cldx$ establecida igual a 1 como entradas, y la agrupación $resSamplesCb$ como salida.

- Cuando $treeType$ es igual a ÁRBOL_ÚNICO, el proceso de decodificación para la señal residual de bloques de codificación codificados en modo de inter-predicción es consultado con la ubicación (x_{Tb0} , y_{Tb0}) establecida igual a la ubicación de croma ($x_{Cb} / SubWidthC$, $y_{Cb} / SubHeightC$), la anchura $nTbW$ establecida igual a la anchura del bloque de codificación de croma $cbWidth / SubWidthC$, la altura $nTbH$ establecida igual a la altura del bloque de codificación de croma $cbHeight / SubHeightC$ y la variable $cldx$ establecida igual a 2 como entradas, y la agrupación $resSamplesCr$ como salida.

4. Las muestras reconstruidas de la unidad de codificación actual son derivadas de la siguiente manera:

- El proceso de reconstrucción de imagen para un componente de color es consultado con la ubicación de bloque (x_{Curr} , y_{Curr}) establecida igual a (x_{Cb} , y_{Cb}), la anchura de bloque $nCurrSw$ establecida igual a $cbWidth$, la altura de bloque $nCurrSh$ establecida igual a $cbHeight$, la variable $cldx$ establecida igual a 0, la agrupación $predSamples(cbWidth) \times (cbHeight)$ establecida igual a $predSamplesL$ y la agrupación $resSamples(cbWidth) \times (cbHeight)$ establecida igual a $resSamplesL$ como entradas, y la salida es una imagen reconstruida modificada antes de la filtración en-bucle.

- Cuando $treeType$ es igual a ÁRBOL_ÚNICO, el proceso de reconstrucción de imagen para un componente de color es consultado con la ubicación de bloque (x_{Curr} , y_{Curr}) establecida igual a ($x_{Cb} / SubWidthC$, $y_{Cb} / SubHeightC$), la anchura de bloque $nCurrSw$ establecida igual a $cbWidth / SubWidthC$, la altura de bloque $nCurrSh$ establecida igual a $cbHeight / SubHeightC$, la variable $cldx$ establecida igual a 1, la agrupación $resSamples(cbWidth / SubWidthC) \times (cbHeight / SubHeightC)$ establecida igual a $predSamplesCb$ y la agrupación $resSamples(cbWidth / SubWidthC) \times (cbHeight / SubHeightC)$ establecida igual a $resSamplesCb$ como entradas, y la salida es una imagen reconstruida modificada antes de la filtración en bucle.

- Cuando $treeType$ es igual a ÁRBOL_ÚNICO, el proceso de reconstrucción de imagen para un

componente de color es consultado con la ubicación de bloque (xCurr, yCurr) establecida igual a (xCb / SubWidthC, yCb / SubHeightC), la anchura de bloque nCurrSw establecida igual a cbWidth / SubWidthC, la altura de bloque nCurrSh establecida igual a cbHeight / SubHeightC, la variable cldx establecida igual a 2, la agrupación predSamples (cbWidth / SubWidthC)x(cbHeight / SubHeightC) establecida igual a predSamplesCr y la agrupación resSamples (cbWidth / SubWidthC)x(cbHeight / SubHeightC) establecida igual a resSamplesCr como entradas, y la salida es una imagen reconstruida modificada antes de la filtración en bucle.

8.6.2 Proceso de derivación para componentes de vector de bloque para bloques IBC

8.6.2.1 General

Las entradas a este proceso son:

una ubicación de luma (xCb, yCb) de la muestra superior-izquierda del bloque de codificación de luma actual con relación a la muestra de luminancia superior-izquierda de la imagen actual,

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.

Las salidas de este proceso son:

el vector de bloque de luma en una precisión bvL de muestra fraccionaria de 1/16.

El vector mvL de bloque de luma es derivado de la siguiente manera:

- El proceso de derivación para la predicción del vector de bloque de luma IBC conforme a lo especificado en la cláusula 8.6.2.2 es consultado con la ubicación de luma (xCb, yCb), las entradas de variables cbWidth y cbHeight, y siendo la salida el vector bvL de bloque de luma. Cuando general_merge_flag[xCb][yCb] es igual a 0, se aplica lo siguiente:

1. La variable bvd es derivada de la siguiente manera:

$$bvd[0] = MvdL0[xCb][yCb][0] \quad (1112)$$

$$bvd[1] = MvdL0[xCb][yCb][1] \quad (1113)$$

2. El proceso de redondeo para vectores de movimiento es consultado con mvX establecido igual a bvL, rightShift establecido igual a AmvrShift, y leftShift establecido igual a AmvrShift como entradas y el bvL redondeado como salida.

3. El vector bvL de bloque de luma es modificado de la siguiente manera:

$$u[0] = (bvL[0] + bvd[0] + 2^{18}) \% 2^{18} \quad (1114)$$

$$bvL[0] = (u[0] \gg 2^{17}) ? (u[0] - 2^{18}) : u[0] \quad (1115)$$

$$u[1] = (bvL[1] + bvd[1] + 2^{18}) \% 2^{18} \quad (1116)$$

$$bvL[1] = (u[1] \gg 2^{17}) ? (u[1] - 2^{18}) : u[1] \quad (1117)$$

NOTA 1 – Los valores resultantes de bvL[0] y bvL[1] conforme a lo antes especificado siempre estarán en el intervalo de -2^{17} a $2^{17} - 1$, inclusivo.

Cuando IsGt4by4 es igual a VERDADERO, el proceso de actualización para la lista de predictores de vector de bloque basado en el histórico conforme a lo especificado en la cláusula 8.6.2.6 es consultado con vector bvL de bloque de luma.

Un requerimiento de la adecuación de la corriente de bits es que el vector bvL de bloque de luma obedezca a las siguientes restricciones:

CtbSizeY es mayor que o igual a $((yCb + (bvL[1] \gg 4)) \& (CtbSizeY - 1)) + cbHeight$.

lbcVirBuf[0][(x + (bvL[0] \gg 4)) \& (lbcBufWidthY - 1)][(y + (bvL[1] \gg 4)) \& (CtbSizeY - 1)] no será igual a -1 para $x = xCb..xCb + cbWidth - 1$, $y = yCb..yCb + cbHeight - 1$.

8.6.2.2 Proceso de derivación para la predicción de vector de bloque de luma IBC

Este proceso solamente es consultado cuando $CuPredMode[0][x_{Cb}][y_{Cb}]$ es igual a $MODE_IBC$, donde (x_{Cb} , y_{Cb}) especifica la muestra superior-izquierda del bloque de codificación de luma actual con relación a la muestra de luma superior-izquierda de la imagen actual.

Las entradas a este proceso son:

10 una ubicación de luma (x_{Cb} , y_{Cb}) de la muestra superior-izquierda del bloque de codificación de luma actual con relación a la muestra de luminancia superior-izquierda de la imagen actual,

una variable $cbWidth$ que especifica la anchura del bloque de codificación actual en muestras de luma,

15 una variable $cbHeight$ que especifica la altura del bloque de codificación actual en muestras de luma.

Las salidas de este proceso son:

El vector bvL de bloque de luma en precisión de muestra fraccionaria de 1/16.

20 El vector bvL de bloque de luma es derivado a través de las siguientes etapas ordenadas:

1. Cuando $IsGt4by4$ es igual a VERDADERO, el proceso de derivación para candidatos de vector de bloque espacial de unidades de codificación contiguas según lo especificado en la cláusula 8.6.2.3 es consultado con la ubicación del bloque de codificación de luma (x_{Cb} , y_{Cb}), la anchura $cbWidth$ y la altura $cbHeight$ del bloque de codificación de luma como entradas, y siendo las salidas los indicadores de disponibilidad $availableFlagA1$, $availableFlagB1$ y los vectores de bloque $bvA1$ y $bvB1$.

2. Cuando $IsGt4by4$ es igual a VERDADERO, la lista de candidatos de vector de bloque, $bvCandList$, es construida de las siguientes maneras:

$i = 0$

si($IndicadorA1disponible$)

35 $bvCandList[i++] = bvA1$ (1118)

si($IndicadorB1disponible$)

40 $bvCandList[i++] = bvB1$

3. La variable $numCurrCand$ es derivada de la siguiente manera:

45 $IsGt4by4$ es igual a VERDADERO, $numCurrCand$ se establece igual al número de candidatos de fusión en la $bvCandList$. De otra manera ($IsGt4by4$ es igual a FALSO), $numCurrCand$ se establece igual a 0.

4. Cuando $numCurrCand$ es menor que $MaxNumIbcMergeCand$ y $NumHmvpIbcCand$ es mayor que 0, el proceso de derivación de candidatos de vector de movimiento basado en el histórico IBC según lo especificado en 8.6.2.4 es consultado con $bvCandList$, y $numCurrCand$ como entradas, y $bvCandList$ y $numCurrCand$ modificadas como salidas.

5. Cuando $numCurrCand$ es menor que $MaxNumIbcMergeCand$, se aplica lo siguiente hasta que $numCurrCand$ es igual a $MaxNumIbcMergeCand$:

55 $bvCandList[numCurrCand][0]$ se establece igual a 0.

$bvCandList[numCurrCand][1]$ se establece igual a 0.

$numCurrCand$ se incrementa en 1.

60 6. La variable $bvIdx$ es derivada de la siguiente manera:

$bvIdx = general_merge_flag[x_{Cb}][y_{Cb}] ? merge_idx[x_{Cb}][y_{Cb}] : mvp_10_flag[x_{Cb}][y_{Cb}]$ (1119)

65 7. Se realizan las siguientes asignaciones:

$$bvL[0] = bvCandList[mvldx][0] \quad (1120)$$

$$bvL[1] = bvCandList[mvldx][1] \quad (1121)$$

5 8.6.2.3 Proceso de derivación para candidatos de vector de bloque espacial IBC

Las entradas a este proceso son:

10 una ubicación de luma (xCb, yCb) de la muestra superior-izquierda del bloque de codificación de luma actual con relación a la muestra de luminancia superior-izquierda de la imagen actual,

una variable cb Width que especifica la anchura del bloque de codificación actual en muestras de luma,

15 una variable cbHeight que especifica la altura del bloque de codificación actual en muestras de luma.

Las salidas de este proceso son de la siguiente manera:

los indicadores de disponibilidad availableFlagA1 y availableFlagB1 de las unidades de codificación contiguas,

20 los vectores de bloque en precisión bvA1, y bvB1 de muestra fraccionaria de 1/16 de las unidades de codificación contiguas.

Para la derivación de IndicadorA1disponible y mvA1 aplica lo siguiente:

25 La ubicación de luma (xNbA1, yNbA1) dentro del bloque de codificación de luma contiguo se establece igual a (xCb - 1, yCb + cbHeight - 1).

30 El proceso de derivación para disponibilidad de bloque contiguo es consultado con la ubicación de luma actual (xCurr, yCurr) establecida igual a (xCb, yCb), la ubicación de luminancia contigua (xNbA1, yNbA1), checkPredModeY establecida igual a VERDADERO, y cldx establecida igual a 0 como entradas, y la salida es asignada al indicador de disponibilidad de bloque availableA1.

Las variables IndicadorA1disponible y bvA1 son derivadas de la siguiente manera:

35 Si A1disponible es igual a FALSO, IndicadorA1disponible se establece igual a 0 y ambos componentes de bvA1 son establecidos iguales a 0. De otra manera, IndicadorA1disponible se establece igual a 1 y se realizan las siguientes asignaciones:

$$bvA1 = MvL0[xNbA1][yNbA1] \quad (1122)$$

40 Para la derivación de IndicadorB1disponible y bvB1 se aplica lo siguiente:

La ubicación de luma (xNbB1, yNbB1) dentro del bloque de codificación de luminancia contiguo se establece igual a (xCb + cbWidth - 1, yCb - 1).

45 El proceso de derivación para disponibilidad de bloque contiguo es consultado con la ubicación de luma actual (xCurr, yCurr) establecida igual a (xCb, yCb), la ubicación de luminancia contigua (xNbB1, yNbB1), checkPredModeY establecida igual a VERDADERO, y cldx establecido igual a 0 como entradas, y la salida es asignada al indicador de disponibilidad de bloque availableB1.

50 Las variables IndicadorB1disponible y bvB1 son derivadas de la siguiente manera:

Si una o más de las siguientes condiciones son verdaderas, IndicadorB1disponible se establece igual a 0 y ambos componentes de bvB1 son establecidos iguales a 0:

55 B1disponible es igual a FALSO.

A1disponible es igual a VERDADERO y las ubicaciones de luma (xNbA1, yNbA1) y (xNbB1, yNbB1) tienen los mismos vectores de bloque.

60 De otra manera, IndicadorB1disponible se establece igual a 1 y se realizan las siguientes asignaciones:

$$bvB1 = MvL0[xNbB1][yNbB1] \quad (1123).$$

65 8.6.2.4 Proceso de derivación para candidatos de vector de movimiento basado en el histórico IBC

Las entradas a este proceso son:

una lista de candidatos de vector de bloque bvCandList,

5 el número de candidatos de vector de bloque disponibles en la lista numCurrCand.

Las salidas a este proceso son:

la lista de candidatos de vector de bloque modificada bvCandList,

10 el número modificado de candidatos de vector de movimiento en la lista numCurrCand.

Las variables isPrunedA1 e isPrunedB1 se establecen igual a FALSO.

15 Para cada candidato en HmvplbcCandList[hMvpIdx] con índice hMvpIdx = 1..NumHmvplbcCand, se repiten las siguientes etapas ordenadas hasta que numCurrCand es igual a MaxNumIbcMergeCand:

1. La variable sameMotion es derivada de la siguiente manera:

20 Si todas las siguientes condiciones son verdaderas para cualquier candidato N de vector de bloque siendo N A1 o B1, sameMotion e isPrunedN son establecidos iguales a VERDADERO:

IsGt4by4 es igual a VERDADERO.

25 hMvpIdx es igual a 1.

El candidato HmvplbcCandList[NumHmvplbcCand – hMvpIdx] es igual al candidato N de vector de bloque.

isPrunedN es igual a FALSO.

30 De otra manera, sameMotion se establece igual a FALSO.

2. Cuando sameMotion es igual a FALSO, el candidato HmvplbcCandList[NumHmvplbcCand – hMvpIdx] se añade a la lista de candidatos de vector de bloque de la siguiente manera:

35
$$bvCandList[numCurrCand++] = HmvplbcCandList[NumHmvplbcCand - hMvpIdx] \quad (1124).$$

8.6.2.5 Proceso de derivación para vectores de bloque de croma

40 La entrada a este proceso es:

un vector de bloque de luma en precisión de muestra fraccionaria de 1/16 bvL.

La salida de este proceso es un vector de bloque de croma en precisión de muestra fraccionaria de 1/32 bvC.

45 Un vector de bloque de croma es derivado a partir del vector de bloque de luma correspondiente.

El vector de bloque de croma bvC es derivado de la siguiente manera:

50
$$bvC[0] = ((bvL[0] >> (3 + SubWidthC)) * 32 \quad (1125)$$

$$bvC[1] = ((bvL[1] >> (3 + SubHeightC)) * 32 \quad (1126)$$

55 8.6.2.6 Proceso de actualización para la lista de candidatos de predictor de vector de bloque basado en el histórico

Las entradas a este proceso son:

vector de bloque de luma bvL en precisión de muestra fraccionaria de 1/16.

60 La lista de candidatos HmvplbcCandList es modificada por las siguientes etapas en orden:

1. La variable identicalCandExist se establece igual a FALSO y la variable removeldx se establece igual a 0.

65 2. Cuando NumHmvplbcCand es mayor que 0, para cada índice hMvpIdx con hMvpIdx = 0..NumHmvplbcCand – 1, se aplican las siguientes etapas hasta que identicalCandExist es igual a

VERDADERO:

Cuando bvL es igual a HmvplbcCandList[hMvpldx], identicalCandExist se establece igual a VERDADERO y removeldx se establece igual a hMvpldx.

5

3. La lista de candidatos HmvplbcCandList es actualizada de la siguiente manera:

Si identicalCandExist es igual a VERDADERO o NumHmvplbcCand es igual a 5, se aplica lo siguiente:

10 Para cada índice i con $i = (\text{removeldx} + 1) \dots (\text{NumHmvplbcCand} - 1)$, HmvplbcCandList[i - 1] se establece igual a HmvplbcCandList[i].

HmvplbcCandList[NumHmvplbcCand - 1] se establece igual a bvL.

15 De otra manera (identicalCandExist es igual a FALSO y NumHmvplbcCand es menor que 5), se aplica lo siguiente:

HmvplbcCandList[NumHmvplbcCand ++] se establece igual a bvL.

20 8.6.3 Proceso de decodificación para bloques IBC

8.6.3.1 General

25 Este proceso es consultado cuando se decodifica una unidad de codificación codificada en modo de predicción IBC.

Las entradas a este proceso son:

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

una variable cb Width que especifica la anchura del bloque de codificación actual en muestras de luma,

35 una variable cbHeight que especifica la altura del bloque de codificación actual en muestras de luma,

el vector de bloque bv,

una variable cldx que especifica el índice de componente de color del bloque actual.

40 Las salidas de este proceso son:

Una agrupación predSamples de muestras de predicción.

45 Cuando cldx es igual a 0, para $x = \text{xCb} \dots \text{xCb} + \text{cbWidth} - 1$ e $y = \text{yCb} \dots \text{yCb} + \text{cbHeight} - 1$, se aplica lo siguiente:

$$\text{xVb} = (x + (\text{bv}[0] \gg 4)) \& (\text{lbcBufWidthY} - 1) \quad (1127)$$

$$\text{yVb} = (y + (\text{bv}[1] \gg 4)) \& (\text{CtbSizeY} - 1) \quad (1128)$$

50

$$\text{predSamples}[x][y] = \text{ibcVirBuf}[0][\text{xVb}][\text{yVb}] \quad (1129)$$

Cuando cldx no es igual a 0, para $x = \text{xCb} / \text{subWidthC} \dots \text{xCb} / \text{subWidthC} + \text{cbWidth} / \text{subWidthC} - 1$ e $y = \text{yCb} / \text{subHeightC} \dots \text{yCb} / \text{subHeightC} + \text{cbHeight} / \text{subHeightC} - 1$, se aplica lo siguiente:

55

$$\text{xVb} = (x + (\text{bv}[0] \gg 5)) \& (\text{lbcBufWidthC} - 1) \quad (1130)$$

$$\text{yVb} = (y + (\text{bv}[1] \gg 5)) \& ((\text{CtbSizeY} / \text{subHeightC}) - 1) \quad (1131)$$

60

$$\text{predSamples}[x][y] = \text{ibcVirBuf}[\text{cldx}][\text{xVb}][\text{yVb}] \quad (1132)$$

Cuando cldx es igual a 0, se realizan las siguientes asignaciones para $x = 0 \dots \text{cbWidth} - 1$ e $y = 0 \dots \text{cbHeight} - 1$:

$$\text{MvL0}[\text{xCb} + x][\text{yCb} + y] = \text{bv} \quad (1133)$$

65

$$\text{MvL1}[\text{xCb} + x][\text{yCb} + y] = 0 \quad (1134)$$

$$\begin{aligned} \text{ReflDxL0}[x_{Cb} + x][y_{Cb} + y] &= -1 & (1135) \\ \text{ReflDxL1}[x_{Cb} + x][y_{Cb} + y] &= -1 & (1136) \\ \text{PredFlagL0}[x_{Cb} + x][y_{Cb} + y] &= 0 & (1137) \\ \text{PredFlagL1}[x_{Cb} + x][y_{Cb} + y] &= 0 & (1138) \\ \text{BcWldx}[x_{Cb} + x][y_{Cb} + y] &= 0 & (1139) \end{aligned}$$

Lo siguiente es una explicación de las aplicaciones del método de codificación así como del método de decodificación como se muestra en las realizaciones antes mencionadas, y de un sistema que las utiliza.

La figura 8 es un diagrama de bloques que muestra un sistema 3100 de suministro de contenido para llevar a cabo el servicio de distribución de contenido. Este sistema 3100 de suministro de contenido incluye el dispositivo 3102 de captura, dispositivo terminal 3106, y opcionalmente incluye el dispositivo 3126 de visualización. El dispositivo 3102 de captura se comunica con el dispositivo terminal 3106 sobre el enlace 3104 de comunicación. El enlace de comunicación puede incluir el canal 13 de comunicación antes descrito. El enlace 3104 de comunicación incluye pero no se limita a WIFI, Ethernet, Cable, inalámbrico (3G/4G/5G), USB, o cualquier tipo de combinación de los mismos, o similar.

El dispositivo 3102 de captura genera datos, y puede codificar los datos mediante el método de codificación como se muestra en las realizaciones anteriores. Alternativamente, el dispositivo 3102 de captura puede distribuir los datos a un servidor de transmisión por secuencias (que no se muestra 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, teléfono inteligente o Pad, ordenador u ordenador portátil, sistema de conferencias de video, PDA, dispositivo montado en vehículo, o una combinación de cualquiera de los anteriores, o similar. Por ejemplo, el dispositivo 3102 de captura puede incluir el dispositivo fuente 12 como se describió antes. Cuando los datos incluyen video, el codificador 20 de video incluido en el dispositivo 3102 de captura realmente puede realizar procesamiento de codificación de video. Cuando los datos incluyen audio (es decir, voz), un codificador de audio incluido en el dispositivo 3102 de captura realmente puede realizar procesamiento de codificación de audio. Para algunos escenarios prácticos, el dispositivo 3102 de captura distribuye los datos de audio y video codificados multiplexando juntos a ambos. Para otros escenarios prácticos, por ejemplo en el sistema de videoconferencias, los datos de audio codificados y los datos de video codificados no son multiplexados. El dispositivo 3102 de captura distribuye los datos de audio codificados y los datos de video 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 recuperación de datos, tal como teléfono inteligente o Pad 3108, ordenador u ordenador portátil 3110, grabadora de video de red (NVR)/ grabadora de video digital (DVR) 3112, TV 3114, decodificador (STB) 3116, sistema 3118 de videoconferencia, sistema 3120 de videovigilancia, asistente digital personal (PDA) 3122, dispositivo 3124 montado en vehículo, o una combinación de cualquiera de los anteriores, o similar con la capacidad para decodificar los datos codificados antes mencionados. Por ejemplo, el dispositivo terminal 3106 puede incluir el dispositivo destino 14 como se describió antes. Cuando los datos codificados incluyen video, el decodificador 30 de video incluido en el dispositivo terminal es priorizado para realizar decodificación de video. Cuando los datos codificados incluyen audio, un decodificador de audio incluido en el dispositivo terminal es priorizado para ejecutar procesamiento de decodificación de audio.

Para un dispositivo terminal con su dispositivo de visualización, por ejemplo, teléfono inteligente o Pad 3108, ordenador u ordenador portátil 3110, la grabadora de video de red (NVR)/ grabadora de video digital (DVR) 3112, la TV 3114, el asistente digital personal (PDA) 3122, o el dispositivo 3124 montado en vehículo, el dispositivo terminal puede alimentar los datos decodificados a su dispositivo de visualización. Para un dispositivo terminal que no está equipado con dispositivo de visualización, por ejemplo, STB 3116, sistema 3118 de videoconferencia, o sistema 3120 de videovigilancia, un dispositivo 3126 de visualización externo es puesto en contacto con él para recibir y mostrar los datos decodificados.

Cuando cada dispositivo en este sistema ejecuta codificación o decodificación, se puede utilizar el dispositivo de codificación de imagen o el dispositivo de decodificación de imagen, tal como se muestra en las realizaciones antes mencionadas.

La figura 9 es un diagrama que muestra una estructura de un ejemplo del dispositivo terminal 3106. Después que el dispositivo terminal 3106 recibe la corriente procedente del dispositivo 3102 de captura, la unidad 3202 de procedimiento de protocolo analiza el protocolo de transmisión de la corriente. El protocolo incluye pero no se limita a Protocolo de Transmisión por Secuencias en Tiempo Real (RTSP), Protocolo de Transferencia de

Hiper Texto (HTTP), protocolo de Transmisión por Secuencias HTTP en Vivo (HLS), MPEG-DASH, protocolo de Transporte en Tiempo Real (RTP), Protocolo de Mensajería en Tiempo Real (RTMP), o cualquier tipo de combinación de los mismos, o similar.

- 5 Después que la unidad 3202 de procedimiento de protocolo procesa la corriente, se genera el archivo de corriente. El archivo es emitido a 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 video codificados. Tal como se describió antes, para algunos escenarios prácticos, por ejemplo en el sistema de videoconferencia, los datos de audio codificados y los datos de video codificados no son multiplexados. En esta situación, los
10 datos codificados son transmitidos al decodificador 3206 de video y al decodificador 3208 de audio sin hacerlo a través de la unidad 3204 de demultiplexación.

- Mediante el procesamiento de demultiplexación, se genera la corriente elemental de video (ES), la ES de audio, y opcionalmente el subtítulo. El decodificador 3206 de video, el cual incluye el decodificador 30 de video como se explicó en las realizaciones antes mencionadas, decodifica la ES de video mediante el método de decodificación como se muestra en las realizaciones antes mencionadas para generar el fotograma de video, y alimenta estos datos a la unidad 3212 de sincronización. El decodificador 3208 de audio, decodifica la ES de audio para generar el fotograma de audio, y alimenta estos datos a la unidad 3212 de sincronización. Alternativamente, el fotograma de video puede almacenarla en una memoria tampón (que no se muestra en la figura 9) antes de alimentarla a la unidad 3212 de sincronización. De manera similar, el fotograma de audio puede almacenarla en una memoria tampón (que no se muestra en la figura 9) antes de alimentarla a la unidad 3212 de sincronización.

- La unidad 3212 de sincronización sincroniza el fotograma de video y el fotograma de audio, y suministra el video/audio a un dispositivo 3214 de visualización de video/audio. Por ejemplo, la unidad 3212 de sincronización sincroniza la presentación de la información de video y audio. La información se puede codificar en la sintaxis utilizando marcas de tiempo concernientes a la presentación de datos de audio y visuales codificados y marcas de tiempo concernientes a la entrega de la propia corriente de datos.

- 30 Si el subtítulo está incluido en la corriente, el decodificador 3210 de subtítulo decodifica el subtítulo, y lo sincroniza con el fotograma de video y el fotograma de audio, y suministra el video/audio/subtítulo a un dispositivo 3216 de visualización de video/audio/subtítulo.

- La presente invención no se limita al sistema antes mencionado, y cualquier dispositivo de codificación de imágenes o dispositivo de decodificación de imágenes en las realizaciones antes mencionadas se puede incorporar en otro sistema, por ejemplo, un sistema de automóvil.

- Ejemplo 1. Un método para construir una lista de fusión de candidatos para modo de copia intra-bloque, IBC, comprendiendo el método:

- 40 insertar un vector de bloque de un bloque contiguo a la izquierda de un bloque actual en una lista de fusión inicial del bloque actual (en un ejemplo, la lista de fusión inicial es una lista vacía antes de esta etapa de inserción), cuando el bloque contiguo a la izquierda está disponible y el bloque contiguo a la izquierda está utilizando modo IBC;

- 45 insertar un vector de bloque de un bloque contiguo por encima del bloque actual en la lista de fusión inicial (en un ejemplo, la lista de fusión inicial es una lista vacía antes de esta etapa, o la lista de fusión inicial comprende un vector de bloque de un bloque contiguo a la izquierda de un bloque actual), cuando el bloque contiguo por encima está disponible, el bloque contiguo por encima está utilizando modo IBC y el vector de bloque del bloque contiguo por encima no es el mismo que el vector de bloque del bloque contiguo a la izquierda;

- 50 insertar un vector de bloque de bloque contiguo por encima–derecha del bloque actual en la lista de fusión inicial (en un ejemplo, la lista de fusión inicial es una lista vacía antes de esta etapa, o la lista de fusión inicial comprende un vector de bloque de un bloque contiguo por encima de un bloque actual, o la lista de fusión inicial comprende un vector de bloque de un bloque contiguo a la izquierda de un bloque actual, o la lista de fusión inicial comprende un vector de bloque de un bloque contiguo por encima de un bloque actual y un vector de bloque de un bloque contiguo a la izquierda de un bloque actual), cuando el bloque contiguo por encima a la derecha está disponible y el bloque contiguo por encima a la derecha está utilizando modo IBC.

- 60 Ejemplo 2. El método del ejemplo 1, en donde el método además comprende:

- insertar un vector de bloque de bloque contiguo inferior a la izquierda del bloque actual en la lista de fusión inicial (en un ejemplo, la lista de fusión inicial es una lista vacía antes de esta etapa, o la lista de fusión inicial comprende un vector de bloque de un bloque contiguo por encima de un bloque actual, o la lista de fusión inicial comprende un vector de bloque de un bloque contiguo a la izquierda de un bloque actual, o la lista de fusión inicial comprende un vector de bloque de un bloque contiguo por encima de un bloque actual y un vector

de bloque de un bloque contiguo a la izquierda de un bloque actual, o la lista de fusión inicial comprende un vector de bloque de un bloque contiguo por encima de un bloque actual, un vector de bloque de un bloque contiguo a la izquierda de un bloque actual y un vector de bloque de un bloque contiguo por encima a la derecha), cuando el bloque contiguo inferior a la izquierda está disponible y el bloque contiguo inferior a la izquierda está utilizando modo IBC.

Ejemplo 3. El método del ejemplo 1 o 2, en donde el método comprende además:

insertar un vector de bloque del bloque contiguo por encima a la izquierda del bloque actual en la lista de fusión inicial (en un ejemplo, la lista de fusión inicial es una lista vacía antes de esta etapa, o la lista de fusión inicial comprende un vector de bloque de un bloque contiguo por encima de un bloque actual, o la lista de fusión inicial comprende un vector de bloque de un bloque contiguo a la izquierda de un bloque actual, o la lista de fusión inicial comprende un vector de bloque de un bloque contiguo por encima de un bloque actual y un vector de bloque de un bloque contiguo a la izquierda de un bloque actual, o la lista de fusión inicial comprende un vector de bloque de un bloque contiguo por encima de un bloque actual, un vector de bloque de un bloque contiguo a la izquierda de un bloque actual y un vector de bloque de un bloque contiguo por encima a la derecha, o la lista de fusión inicial comprende un vector de bloque de un bloque contiguo por encima de un bloque actual, un vector de bloque de un bloque contiguo a la izquierda de un bloque actual, un vector de bloque de un bloque contiguo por encima a la derecha y un vector de bloque de un bloque contiguo inferior a la izquierda), cuando el bloque contiguo por encima a la izquierda está disponible, el bloque contiguo por encima a la izquierda está utilizando modo IBC y un número de vectores de bloque en la lista de fusión inicial es más pequeño que un umbral (por ejemplo, el umbral es 4).

Ejemplo 4. Un codificador (20) que comprende circuitos de procesamiento para llevar a cabo el método según cualquiera de los ejemplos 1 a 3.

Ejemplo 5. Un decodificador (30) que comprende circuitos de procesamiento para llevar a cabo el método según cualquiera de los ejemplos 1 a 3.

Ejemplo 6. Un producto de programa informático que comprende código de programa para ejecutar el método según cualquiera de los ejemplos 1 a 3.

Ejemplo 7. Un decodificador o un codificador, que comprende:

uno o más procesadores; y

un medio de almacenamiento legible por ordenador no transitorio acoplado a los procesadores y que almacena programación para ejecución por los procesadores, en donde la programación, cuando es ejecutada por los procesadores, configura el decodificador para llevar a cabo el método según cualquiera de los ejemplos 1 a 3.

Operadores matemáticos

Los operadores matemáticos utilizados en esta solicitud son similares a los utilizados en el lenguaje de programación C. Sin embargo, los resultados de las operaciones de división de número entero y desplazamiento aritmético son definidos de manera más precisa, y se definen operaciones adicionales, tales como exponenciación y división en valor real. Convenciones de numeración y cómputo generalmente comienzan desde 0, por ejemplo, "el primero" es equivalente al 0-ésimo, "el segundo" es equivalente al 1-ésimo, etc.

Operadores aritméticos

A continuación se definen los siguientes operadores aritméticos:

+ Suma

- Resta (como un operador de dos argumentos) o negación (como un operador de prefijo unitario)

* Multiplicación, incluyendo multiplicación matricial

x^y Exponenciación. Especifica x a la potencia de y . En otros contextos, tal notación es utilizada para sobrescribir de forma no pretendida para interpretación como exponenciación.

/ División de número entero 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 utiliza para indicar división en ecuaciones matemáticas donde no se pretende truncamiento o

redondeo.

x/y Utilizado para indicar división en ecuaciones matemáticas donde no se pretende truncamiento o redondeo.

5

$$\sum_{i=x}^y f(i)$$

La suma de f(i), tomando i todos los valores enteros desde x hasta, e incluyendo y

X%y Módulo. Resto de x dividido por y, definido solamente para números enteros x e y con x >= 0 e y > 0.

10 Operadores lógicos

A continuación se definen los siguientes operadores lógicos:

x&&y Lógica Booleana "y" de x e y

15

x || y Lógica Booleana "o" de x e y

! Lógica Booleana "no"

20 x ? y : z Si x es VERDADERO o no igual a 0, evalúa al valor de y; de otra manera, evalúa al valor de z.

Operadores de Relación

A continuación se definen los siguientes operadores de relación:

25

> Mayor que

>= Mayor que o igual a

30

< Menor que

<= Menor que o igual a

= = Igual a

35

!= No igual a

Cuando se aplica un operador de relación a un elemento de sintaxis o variable al que se ha asignado el valor "na" (no aplicable), el valor "na" es tratado como un valor distinto para el elemento de sintaxis o variable. El valor "na" es considerado no igual a ningún otro valor.

40

Operadores a nivel de bit

A continuación se definen los siguientes operadores a nivel de bit:

45

& Nivel de bit "y". Cuando opera en argumentos de número entero, opera en una representación de complemento de dos del valor entero. Cuando opera en un argumento binario que contiene menor cantidad de bits que otro argumento, el argumento más corto es extendido añadiendo bits más significativos iguales a 0.

50

| Nivel de bit "o". Cuando opera en argumentos de número entero, opera en una representación de complemento de dos del valor entero. Cuando opera en un argumento binario que contiene menor cantidad de bits que otro argumento, el argumento más corto es extendido añadiendo bits más significativos iguales a 0.

55

^ Nivel de bits "o exclusivo". Cuando opera en argumentos de número entero, opera en una representación de complemento de dos del valor entero. Cuando opera en un argumento binario que contiene menor cantidad de bits que otro argumento, el argumento más corto es extendido añadiendo bits más significativos iguales a 0.

60

x >> y Desplazamiento aritmético a la derecha de una representación de número entero de complemento de dos de dígitos binarios x por y. Esta función es definida únicamente para valores enteros no negativos de y. Bits desplazados a los bits más significativos (MSB) como resultado del desplazamiento a la derecha tienen un valor igual al MSB de x antes de la operación de desplazamiento.

x << y Desplazamiento aritmético a la izquierda de una representación de número entero de complemento

de dos de dígitos binarios x por y . Esta función es definida únicamente para valores enteros no negativos de y . 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

A continuación se definen los siguientes operadores aritméticos:

10 = Operador de asignación

+ + Incremento, es decir, $x++$ es equivalente a $x = x + 1$; cuando se utiliza en un índice de agrupación, evalúa al valor de la variable antes de la operación de incremento.

15 - - Decremento, es decir, $x--$ es equivalente a $x = x - 1$; cuando se utiliza en un índice de agrupación, evalúa al valor de la variable antes de la operación de decremento.

+ = Incremento por cantidad especificada, es decir, $x += 3$ es equivalente a $x = x + 3$, y
 $x += (-3)$ es equivalente a $x = x + (-3)$.

20 - = Decremento por cantidad especificada, es decir, $x -= 3$ es equivalente a $x = x - 3$, y
 $x -= (-3)$ es equivalente a $x = x - (-3)$.

Anotación de intervalo

25 La siguiente anotación se utiliza para especificar un intervalo de valores:

$x = y..z$ x toma valores enteros comenzando desde y a z , inclusive, siendo x , y , y z números enteros y siendo z mayor que y .

30 Funciones matemáticas

Se definen las siguientes funciones matemáticas:

35
$$\text{Abs}(x) = \begin{cases} x & ; \quad x \geq 0 \\ -x & ; \quad x < 0 \end{cases}$$

$\text{Asin}(x)$ la función trigonométrica inversa del seno, operando en un argumento x que es el intervalo de -1.0 a 1.0 , inclusive, con un valor de salida en el intervalo de $-\pi/2$ a $\pi/2$, inclusive, en unidades de radianes

40 $\text{Atan}(x)$ la función trigonométrica inversa de la tangente, operando en un argumento x , con un valor de salida en el intervalo de $-\pi/2$ a $\pi/2$, inclusivo, en unidades de radianes

$$\text{Atan2}(y, x) = \begin{cases} \text{Atan}\left(\frac{y}{x}\right) & ; \quad x > 0 \\ \text{Atan}\left(\frac{y}{x}\right) + \pi & ; \quad x < 0 \text{ \& \& } y \geq 0 \\ \text{Atan}\left(\frac{y}{x}\right) - \pi & ; \quad x < 0 \text{ \& \& } y < 0 \\ +\frac{\pi}{2} & ; \quad x = 0 \text{ \& \& } y \geq 0 \\ -\frac{\pi}{2} & ; \quad \text{de otra manera} \end{cases}$$

45 $\text{Ceil}(x)$ el entero más pequeño mayor que o igual a x .

$$\text{Clip1}_v(x) = \text{Clip3}(0, (1 \ll \text{BitDepth}_v) - 1, x)$$

$$\text{Clip1}_c(x) = \text{Clip3}(0, (1 \ll \text{BitDepth}_c) - 1, x)$$

50
$$\text{Clip3}(x, y, z) = \begin{cases} x & ; \quad z < x \\ y & ; \quad z > y \\ z & ; \quad \text{de otra manera} \end{cases}$$

$\text{Cos}(x)$ la función trigonométrica del coseno operando en un argumento x en unidades de radianes.

55 $\text{Floor}(x)$ el entero más grande menor que o igual a x .

$$\text{GetCurrMsb}(a, b, c, d) = \begin{cases} c + d & ; \quad b - a \geq d / 2 \\ c - d & ; \quad a - b \geq d / 2 \\ c & ; \quad \text{de otra manera} \end{cases}$$

$\text{Ln}(x)$ el logaritmo natural de x (el algoritmo de base- e , donde e es la constante base de logaritmo natural 2.718 281 828...).

5

$\text{Log2}(x)$ el logaritmo de base 2 de x .

$\text{Log10}(x)$ el logaritmo de base 10 de x .

10

$$\text{Min}(x, y) = \begin{cases} x & ; \quad x \leq y \\ y & ; \quad x > y \end{cases}$$

$$\text{Max}(x, y) = \begin{cases} x & ; \quad x \geq y \\ y & ; \quad x < y \end{cases}$$

$\text{Round}(x) = \text{Sign}(x) * \text{Floor}(\text{Abs}(x) + 0.5)$

15

$$\text{Sign}(x) = \begin{cases} 1 & ; \quad x > 0 \\ 0 & ; \quad x = 0 \\ -1 & ; \quad x < 0 \end{cases}$$

$\text{Sin}(x)$ la función trigonométrica del seno operando en un argumento x en unidades de radianes

20

$$\text{Sqrt}(x) = \sqrt{x}$$

$$\text{Swap}(x, y) = (y, x)$$

$\text{Tan}(x)$ la función trigonométrica de tangente operando en un argumento x en unidades de radianes

25

Orden de precedencia de operación

Cuando un orden de precedencia en una expresión no es indicado explícitamente por el uso de paréntesis, aplican las siguientes reglas:

30

– Operaciones de una precedencia superior son evaluadas antes que cualquier operación de una precedencia inferior.

– Operaciones de la misma precedencia son evaluadas en secuencia de izquierda a derecha.

35

La siguiente tabla especifica la precedencia de las operaciones desde la más alta a la más baja; una posición más elevada en la tabla indica una precedencia superior.

Para aquellos operadores que también son utilizados en el lenguaje de programación C, el orden de precedencia utilizado en esta Memoria Descriptiva es el mismo que el utilizado en el lenguaje C de programación.

40

Tabla: Precedencia de operación desde la más alta (en la parte superior de la tabla) a la más baja (en la parte inferior de la tabla)

45

operaciones (con operandos $x, y, y z$)
" $x++$ ", " $x--$ "
" $!x$ ", " $-x$ " (como un operador de prefijo unitario)
x^y
" $x * y$ ", " x / y ", " $x \div y$ ", " $\frac{x}{y}$ ", " $x \% y$ "
" $x + y$ ", " $x - y$ " (como un operador de dos argumentos), " $\sum_{i=x}^y f(i)$ "
" $x \ll y$ ", " $x \gg y$ "

"x < y", "x <= y", "x > y", "x >= y"
"x == y", "x != y"
"x & y"
"x y"
"x && y"
"x y"
"x ? y : z"
"x..y"
"x = y", "x += y", "x -= y"

Descripción de texto de operaciones lógicas

En el texto, una declaración de operaciones lógicas tal como se describiría matemáticamente en la siguiente forma:

5

si(condición 0)

declaración 0

10

además si (condición 1)

declaración 1

15

...

además /* comentario informativo sobre la condición remanente */

declaración n

20

se puede describir de la siguiente manera:

... como sigue / ... se aplica lo siguiente:

25

– Si condición 0, declaración 0

– De otra manera, si condición 1, declaración 1

– ...

30

– De otra manera (comentario informativo sobre la condición remanente), declaración n

Cada declaración "Si ... De otra manera, si ... De otra manera, ..." en el texto se introduce con "... como sigue" o "... se aplica lo siguiente" inmediatamente seguido por "Si ... ". La última condición de "Si ... De otra manera, si ... De otra manera, ..." siempre es un "De otra manera, ...". Declaraciones intercaladas "Si ... De otra manera, si ... De otra manera, ..." pueden ser identificadas haciendo corresponder "... como sigue" o "... se aplica lo siguiente" finalizando con "De otra manera, ...".

35

En el texto, una declaración de operaciones lógicas tal como se describiría matemáticamente en la siguiente forma:

40

si(condición 0a && condición 0b)

declaración 0

45

además si (condición 1a || condición 1b)

declaración 1

50

...

además

declaración n

5 puede ser descrito de la siguiente manera:

... como sigue / ... se aplica lo siguiente:

10 – Si todas las siguientes condiciones son verdaderas, declaración 0:

– condición 0a

– condición 0b

15 – De otra manera, si una o más de las siguientes condiciones son verdaderas, declaración 1:

– condición 1a

– condición 1b

20

– ...

– De otra manera, declaración n

25 En el texto, una declaración de operaciones lógicas tal como se describiría matemáticamente en la siguiente forma:

si (condición 0)

30 declaración 0

si (condición 1)

declaración 1

35

puede ser descrita de la siguiente manera:

Cuando condición 0, declaración 0

40 Cuando condición 1, declaración 1.

Aunque realizaciones de la invención han sido descritas principalmente basándose en la codificación de video, debe observarse que realizaciones del sistema 10 de codificación, codificador 20 y decodificador 30 (y de manera correspondiente el sistema 10) y las otras realizaciones descritas en la presente memoria también se pueden configurar para procesamiento o codificación de imágenes estáticas, es decir, el procesamiento o codificación de una imagen individual independiente de cualquier imagen precedente o consecutiva como en codificación de video. En general solamente las unidades de inter-predicción 244 (codificador) y 344 (decodificador) pueden no estar disponibles en el caso de que la codificación de procesamiento de imagen esté limitada a una sola imagen 17. Todas las otras funcionalidades (también denominadas como herramientas o tecnologías) del codificador 20 de video y decodificador 30 de video igualmente se pueden utilizar para procesamiento de imágenes estáticas, por ejemplo cálculo residual 204/304, transformada 206, cuantificación 208, cuantificación inversa 210/310, transformada (inversa) 212/312, división en particiones 262/362, intra-predicción 254/354, y/o filtración 220, 320 en bucle, y codificación entrópica 270 y decodificación entrópica 304. Realizaciones, por ejemplo del codificador 20 y del decodificador 30, y funciones descritas en la presente memoria, por ejemplo con referencia al codificador 20 y al decodificador 30, pueden ser implementadas en hardware, software, firmware, o cualquier combinación de los mismos. Si se implementan en software, las funciones pueden ser almacenadas en un medio legible por ordenador o transmitidas sobre el medio de comunicación como una o más instrucciones o código y ejecutadas por una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, lo cual corresponde a un medio tangible tal como medios de almacenamiento de datos, o medios de comunicación incluyendo cualquier medio que facilite la transferencia de un programa informático de un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador generalmente pueden corresponder a (1) medio de almacenamiento legible por ordenador tangible que es no-transitorio o (2) un medio de comunicación tal como una señal u onda portadora. Los medios de almacenamiento de datos pueden ser cualesquiera medios disponibles a los que se pueda tener acceso a través de uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o

estructuras de datos para implementación de las técnicas descritas en esta descripción. Un producto de programa informático puede incluir un medio legible por ordenador.

5 A manera de ejemplo, y no limitación, dichos medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento de disco óptico, almacenamiento de disco magnético, otros dispositivos de almacenamiento magnético, memoria flash, o cualquier otro medio que pueda ser utilizado para almacenar un código de programa deseado en la forma de instrucciones o estructuras de datos y a las que se puede tener acceso a través de un ordenador. También, cualquier conexión es determinada de manera apropiada un medio legible por ordenador. Por ejemplo si las instrucciones son transmitidas desde un sitio Web, servidor, u otra fuente remota utilizando un cable coaxial, cable de fibra óptica, par trenzado, línea de suscriptor digital (DSL), o tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, cable de fibra óptica, el par trenzado, DSL, o tecnologías inalámbricas tales como infrarrojos, radio, y microondas son incluidas en la definición 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 de portadora, señales u otros medios transitorios, sino que más bien están dirigidos a medios de almacenamiento tangibles no transitorios. Disco (disk) y disco (disc), tal como aquí se utilizan, incluyen disco compacto (CD), disco láser, disco óptico, disco versátil digital (DVD), disquete y disco Blu-ray, donde los discos (disks) generalmente reproducen datos de forma magnética, mientras que los discos (discs) reproducen datos de manera óptica con láseres. Combinaciones de lo anterior también se deberían incluir dentro del alcance de medios legibles por ordenador.

Las instrucciones pueden ser ejecutadas por uno o más procesadores, tal como uno o más procesadores de señal digital (DSP), microprocesadores de propósito general, circuitos integrados de aplicación específica (ASIC), matrices lógicas programables en campo (FPGA), u otros circuitos lógicos discretos o integrados equivalentes. Por consiguiente, el término "procesador" como se utiliza en la presente memoria se puede referir a cualquiera de la estructura anterior o a cualquier otra estructura adecuada para implementación de las técnicas descritas en la presente memoria. Además, en algunos aspectos, la funcionalidad aquí descrita puede ser proporcionada dentro de módulos de hardware y/o software dedicados configurados para codificación y decodificación, o incorporados en un códec combinado. También, las técnicas podrían ser implementadas completamente en uno o más circuitos o elementos lógicos.

Las técnicas de esta descripción pueden ser implementadas en una amplia variedad de dispositivos o aparatos, incluyendo un equipo inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). Varios componentes, módulos o unidades se describen en esta descripción para enfatizar aspectos funcionales de dispositivos configurados para ejecutar las técnicas descritas, pero no necesariamente requieren la realización mediante diferentes unidades de hardware. Más bien, como se describió antes, varias unidades se pueden combinar en una unidad de hardware de códec o pueden ser proporcionadas por un agrupamiento de unidades de hardware inter-operativas, incluyendo uno o más procesadores como se describió antes, en combinación con software y/o firmware adecuado.

REIVINDICACIONES

1. Un método para construir una lista de fusión de candidatos para modo de copia Intra-bloques, IBC, caracterizado por que comprende:
 - 5 insertar un vector de bloque de un bloque contiguo a la izquierda de un bloque actual en una lista de fusión inicial del bloque actual, cuando el bloque contiguo a la izquierda está disponible y el bloque contiguo a la izquierda está utilizando modo IBC;
 - 10 insertar un vector de bloque de un bloque contiguo por encima del bloque actual en la lista de fusión inicial, cuando el bloque contiguo por encima está disponible, el bloque contiguo por encima está utilizando modo IBC y el vector de bloque del bloque contiguo por encima no es el mismo que el vector de bloque del bloque contiguo a la izquierda;
 - 15 insertar un vector de bloque del último candidato en una lista de predictores de vector de movimiento basado en el histórico, HMVP en la lista de fusión inicial, cuando el vector de bloque del bloque contiguo por encima no es el mismo que el vector de bloque del último candidato en la lista de HMVP y cuando el vector de bloque del bloque contiguo a la izquierda no es el mismo que el vector de bloque del último candidato de la lista de HMVP;
 - 20 insertar uno por uno vectores de bloque de otros candidatos en la lista de HMVP en la lista de candidatos de fusión sin recortar, hasta que un número de vectores de bloque en la lista de candidatos de fusión sea igual a un número máximo de vectores de bloque en la lista de candidatos de fusión o hasta que los vectores de bloque de todos los candidatos en la lista de HMVP estén insertados en la lista de candidatos de fusión, en donde los medios de recorte comparan si dos candidatos de fusión en la lista de candidatos de fusión son el mismo,
 - 25 en donde si se alcanza el número máximo de vectores de bloque en la lista de candidatos de fusión, el último vector de bloque en la lista de candidatos de fusión, el último vector de bloque en la lista de candidatos de fusión del bloque actual es un vector de bloque de uno de los otros candidatos de la lista HMVP; y
 - 30 en donde si todos los candidatos de la lista de HMVP se insertan en la lista de candidatos de fusión y el número de vectores de bloque de la lista de candidatos de fusión es menor que el número máximo de vectores de bloque de la lista de candidatos de fusión, se insertan uno o más ceros en la lista de candidatos de fusión hasta que el número de vectores de bloque en la lista de candidatos de fusión es igual al número máximo de vectores de bloque en la lista de candidatos de fusión.
 - 35
2. Un decodificador (30) que comprende circuitos de procesamiento para llevar a cabo el método según la reivindicación 1.
- 40 3. Un producto de programa informático que comprende un código de programa para realizar el método según la reivindicación 1.

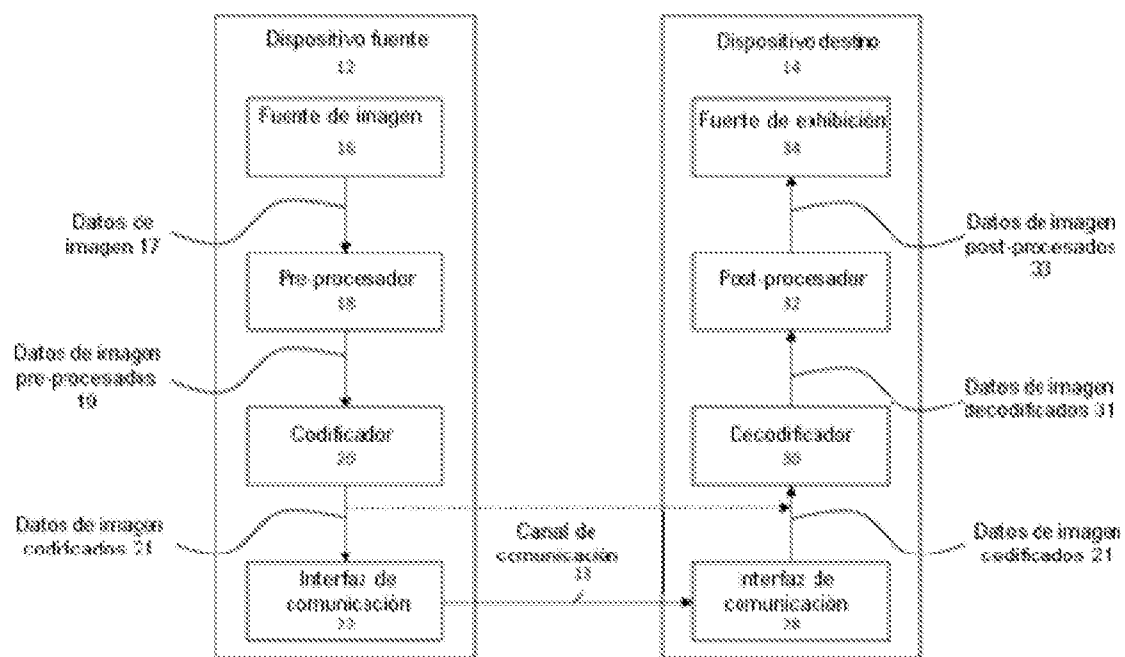


FIGURA 1A

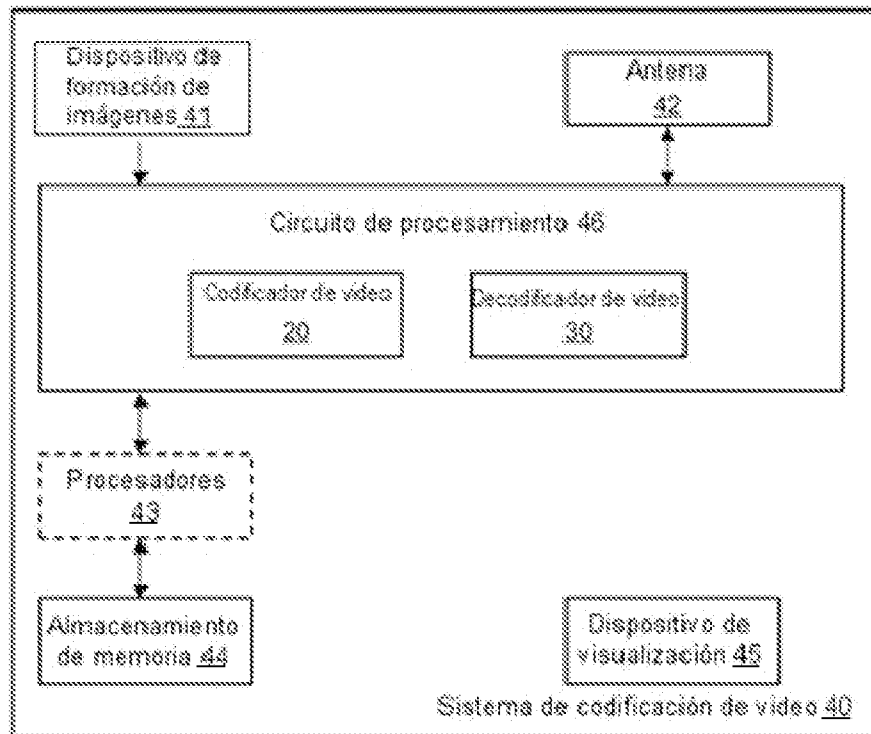


FIGURA 1B

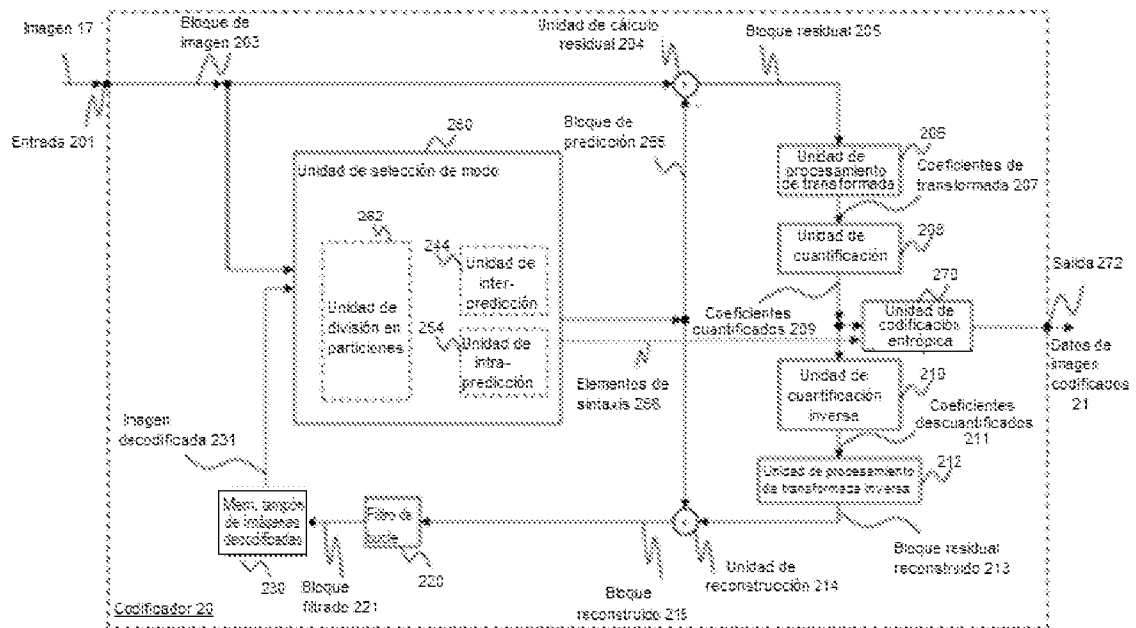
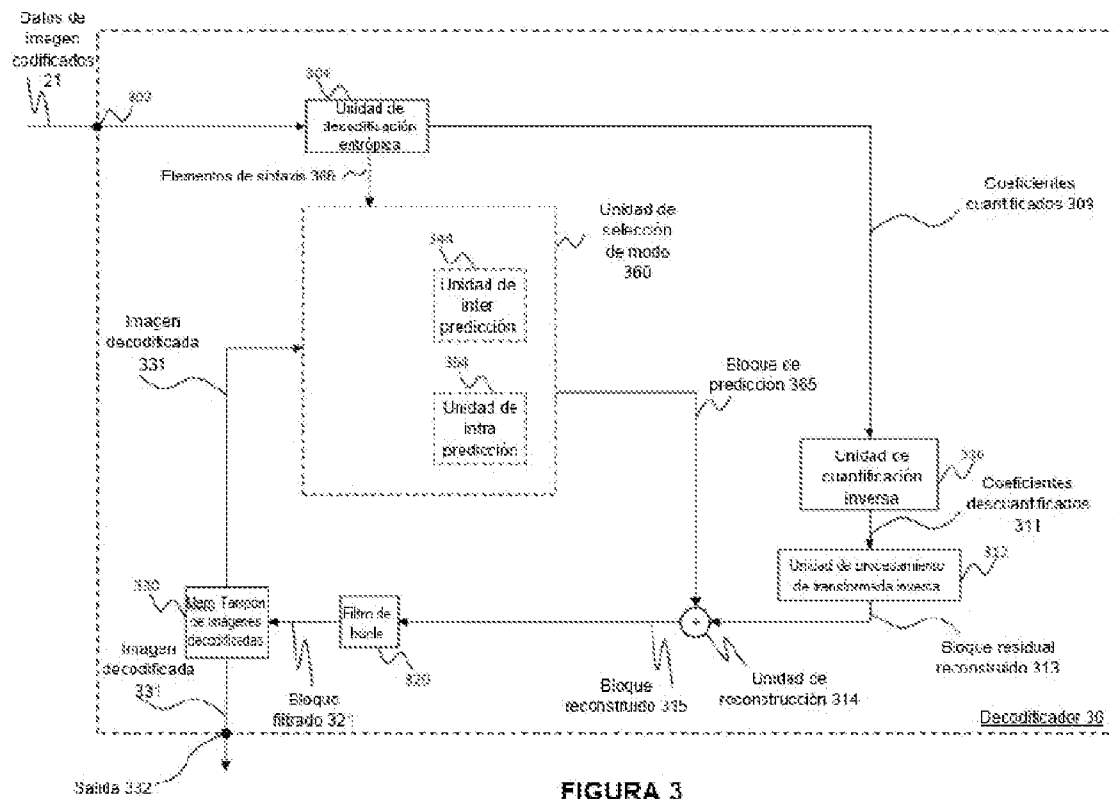


FIGURA 2



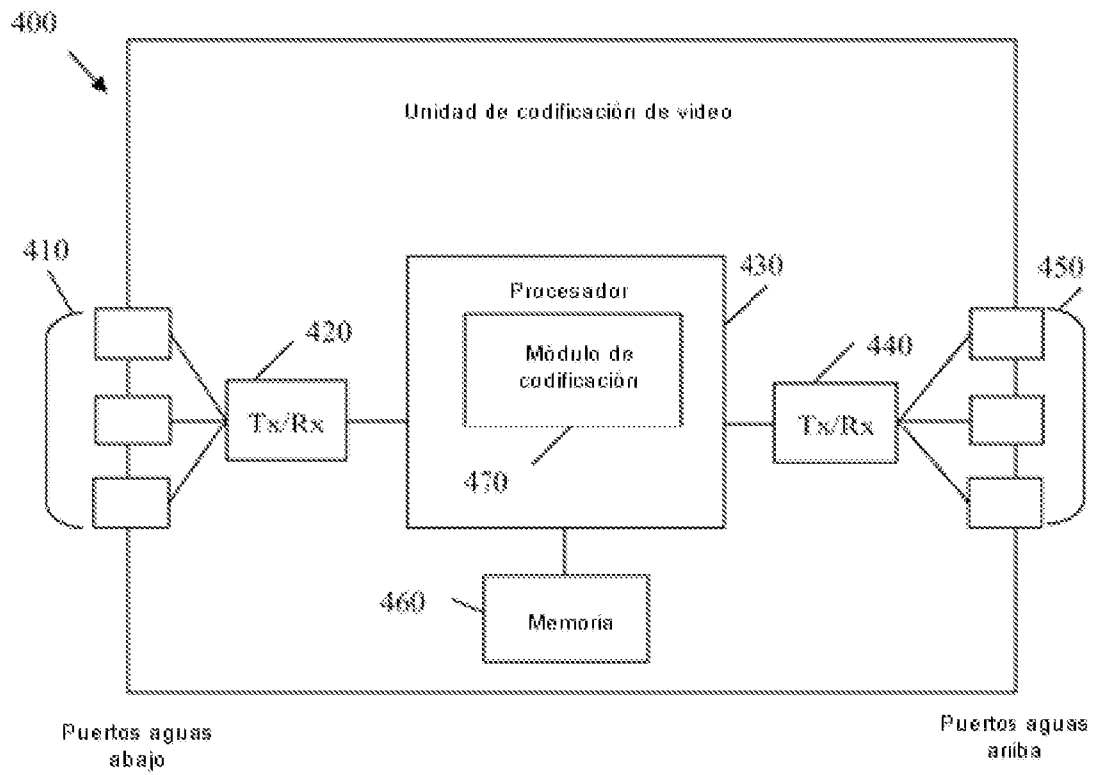


FIGURA 4

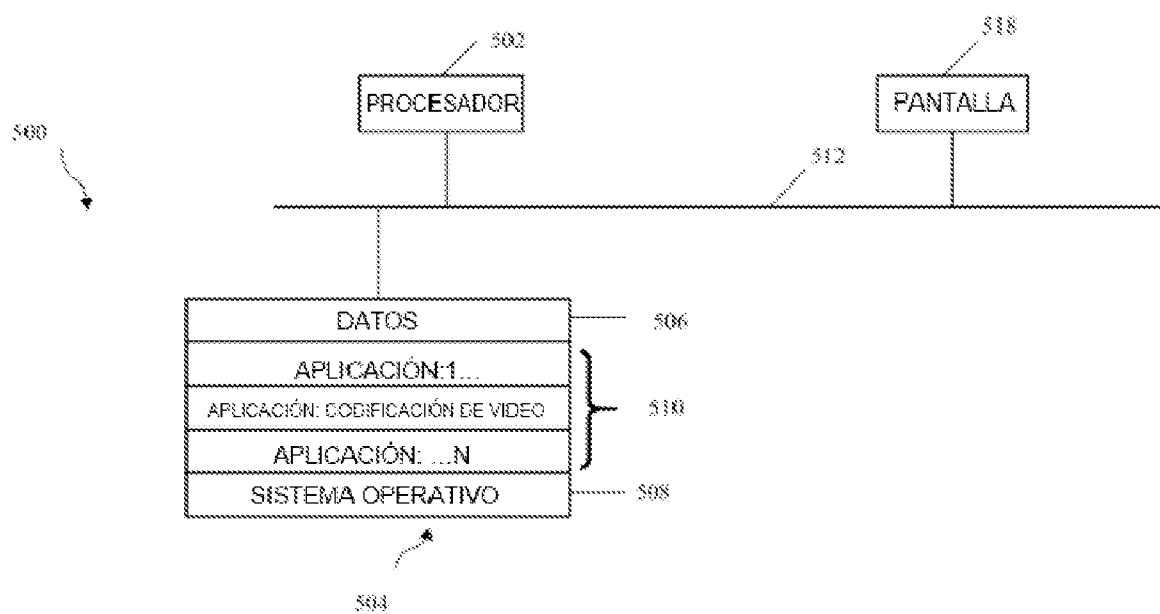


FIGURA 5

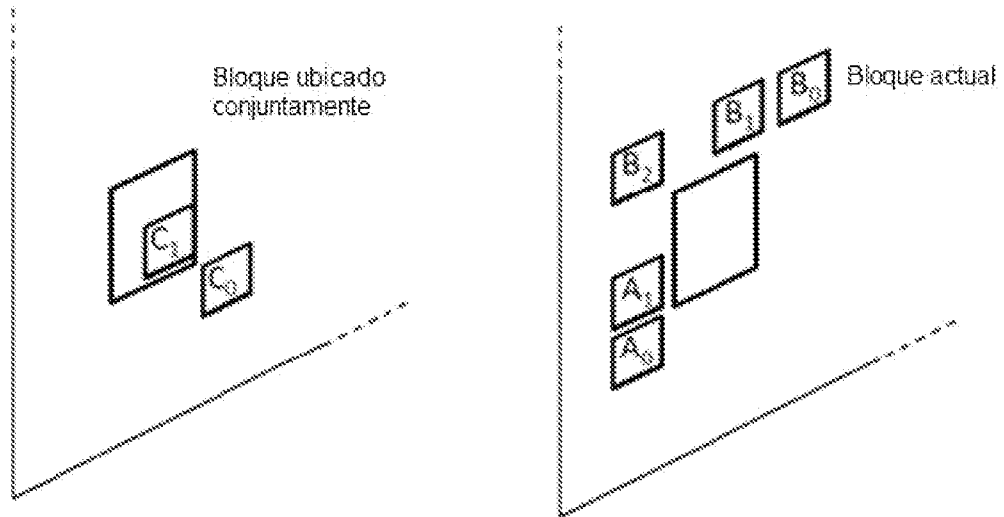


FIGURA 6

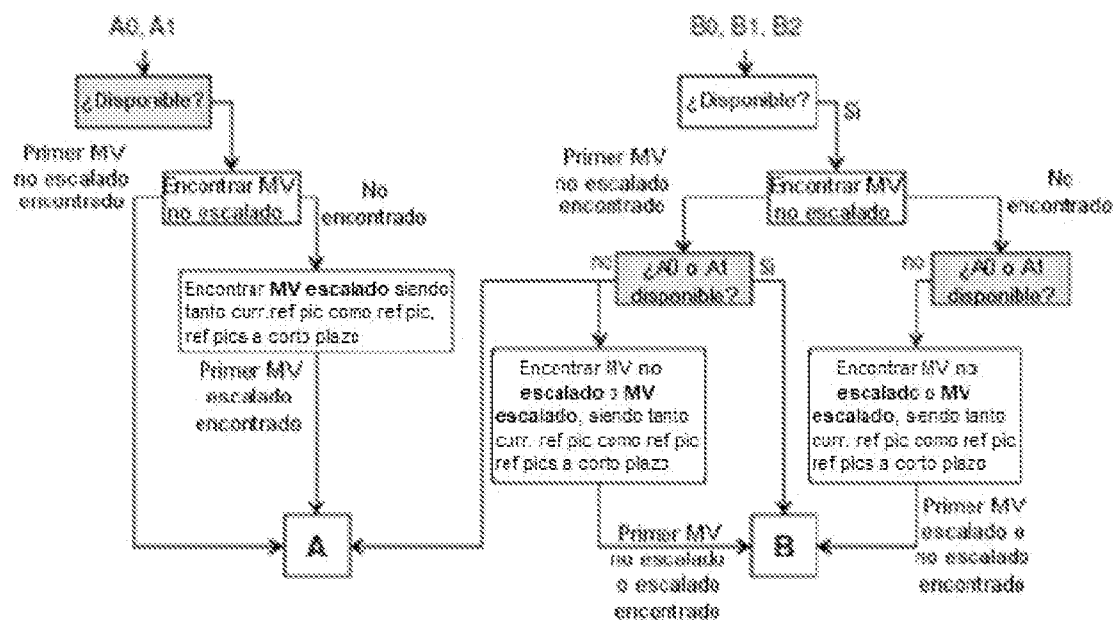


FIGURA 7

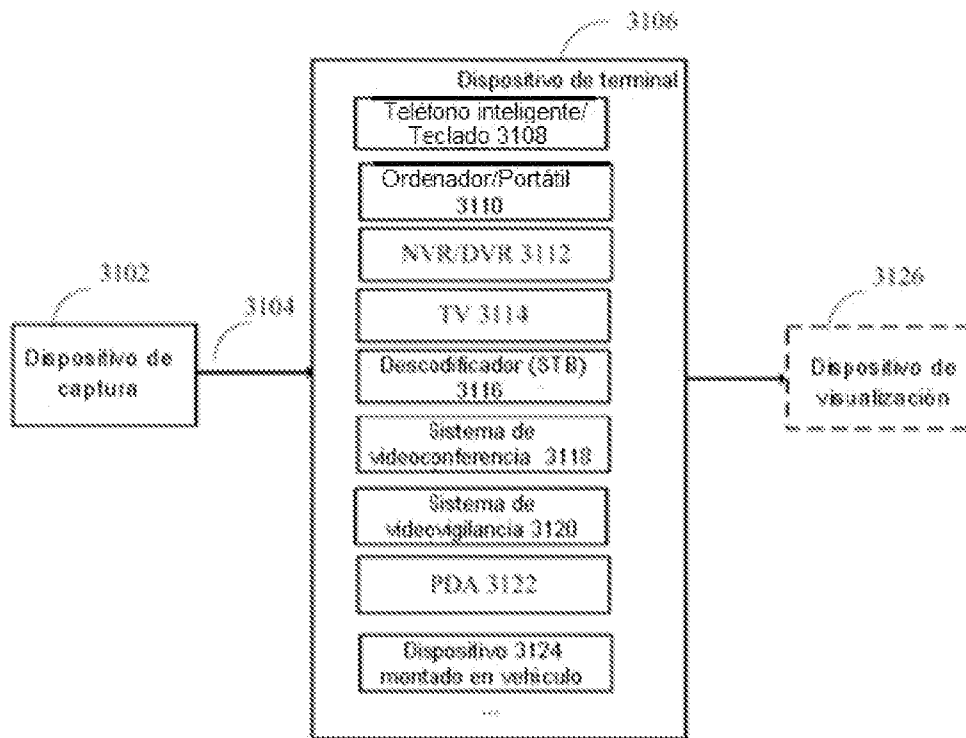


FIGURA 8

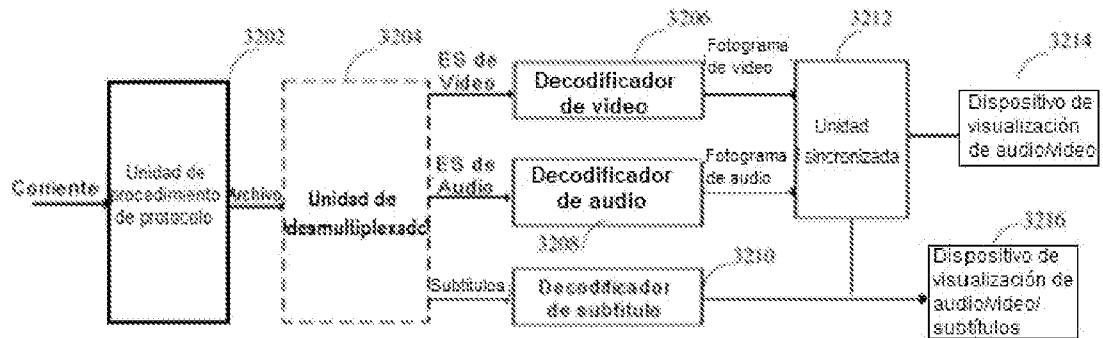


FIGURA 9