

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 928 248**

51 Int. Cl.:

**H04N 19/17** (2014.01)

**H04N 19/174** (2014.01)

**H04N 19/176** (2014.01)

**H04N 19/436** (2014.01)

**H04N 19/70** (2014.01)

12

## TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **01.11.2012 PCT/US2012/063027**

87 Fecha y número de publicación internacional: **10.05.2013 WO13067158**

96 Fecha de presentación y número de la solicitud europea: **01.11.2012 E 12790745 (9)**

97 Fecha y número de publicación de la concesión europea: **21.09.2022 EP 2774369**

54 Título: **Relleno de segmentos en unidades NAL de la porción codificada**

30 Prioridad:

**04.11.2011 US 201161555932 P**

**08.11.2011 US 201161557259 P**

**13.07.2012 US 201213548825**

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

**16.11.2022**

73 Titular/es:

**QUALCOMM INCORPORATED (100.0%)**

**5775 Morehouse Drive**

**San Diego, CA 92121-1714, US**

72 Inventor/es:

**WANG, YE-KUI;**

**COBAN, MUHAMMED ZEYD y**

**KARCZEWICZ, MARTA**

74 Agente/Representante:

**ISERN JARA, Jorge**

ES 2 928 248 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

Relleno de segmentos en unidades NAL de la porción codificada

## 5 Campo técnico

Esta divulgación se refiere a la codificación de video (es decir, la codificación o decodificación de datos de video).

## 10 Antecedentes

Las capacidades de video digital se pueden incorporar en una amplia gama de dispositivos, incluidos televisores digitales, sistemas de transmisión directa digital, sistemas de transmisión inalámbricos, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos celulares o de radio satelital, dispositivos de videoconferencia y similares. Los dispositivos de video digital implementan técnicas de compresión de video, tal como aquellas descritas en los estándares definidos por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Part 10, Codificación avanzada de video (AVC), el estándar de codificación de video de alta eficiencia (HEVC) actualmente en desarrollo, y extensiones de dichos estándares, para transmitir, recibir y almacenar información de video digital de manera más eficiente.

Las técnicas de compresión de video realizan predicción espacial (dentro de la imagen) y/o predicción temporal (entre las imágenes) para reducir o eliminar la redundancia inherente a las secuencias de video. Para la codificación de video basada en bloques, un segmento de video puede dividirse en bloques de video, que también pueden denominarse bloques de árbol, unidades de codificación (CU) y/o nodos de codificación. Los bloques de video en una porción intracodificada (I) de una imagen se codifican usando predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen. Los bloques de video en una porción intercodificada (P o B) de una imagen pueden usar predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen o predicción temporal con respecto a muestras de referencia en otras imágenes de referencia. Las imágenes pueden denominarse marcos, y las imágenes de referencia pueden denominarse marcos de referencia.

"WD4: Borrador de Trabajo 4 de Codificación de Video de Alta Eficiencia" (JCTVC-F803\_d0) de Bross et al., Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, sexta reunión: Torino, IT, 14-22 de julio de 2011 describe un borrador de trabajo del proyecto HEVC que data de julio de 2011.

En "Comments on H.26L NAL Design" (VCEGL26) de Thomas Stockhammer, ITU-VCEG, décimo segunda reunión, Eibsee, Alemania, enero de 2001, se divulga una propuesta para el diseño NAL relativa a su interfaz con una capa de codificación de video.

En el documento WO 2010/050157 A1, se divulga un método de codificación de imágenes, que puede homogeneizar la calidad de imagen de una imagen en su conjunto sin reducir la eficiencia de codificación, que se puede operar a alta tasa mediante la reproducción aleatoria de macrobloques sin cambiar la estructura de la porción.

En "Tiles" de Fuldseth et al., (JCTVC-F335), Joint Collaborative Team on Video Coding (JCT-VC) of ITUT SG16 WP3 and ISO/IEC JTC1/SC29/WG11, sexta reunión: Torino, IT, del 14 al 22 de julio de 2011, se divulga un perfeccionamiento de una técnica de codificación llamada Tiles que particiona una imagen en segmentos rectangulares.

En "Wavefront and Cabac Flush: Different Degrees of Parallelism Without Transcoding" de Clare et al. (JCTVC-F275), se propone combinar WPP con un flush y reinicialización de las variables de estado internas de CABAC al final de cada línea de LCU. Se afirma que la conversión de un flujo de bits de un nivel dado de paralelismo a otro se puede lograr proporcionando puntos de entrada como mensajes SEI.

## Resumen

En general, esta divulgación describe técnicas para codificar y decodificar datos de video. Un codificador de video puede dividir una imagen en una pluralidad de particiones de imagen. Las particiones de imagen incluyen subconjuntos no superpuestos de los bloques de árbol de la imagen. Los tipos de ejemplo de particiones de imagen incluyen mosaicos y ondas de procesamiento paralelo de frente de onda (WPP). De acuerdo con la invención, el codificador de video genera una unidad de capa de abstracción de red (NAL) de porción codificada que incluye representaciones codificadas de los bloques de árbol asociados con una porción de la imagen. El codificador de video genera la unidad NAL de la porción codificada de tal manera que los bloques de árbol codificados se agrupan dentro de la unidad NAL de la porción codificada por las particiones de imagen a las que pertenecen los bloques de árbol. El codificador de video rellena uno o más de los segmentos de modo que cada uno de los segmentos comience en un límite de byte. Un decodificador de video puede decodificar bloques de árbol codificados de la unidad NAL de porción codificado.

La invención está definida por las reivindicaciones independientes. Los ejemplos presentados en esta divulgación que, por sí mismos, no incluyen todas las características de las reivindicaciones independientes, no representan, por sí mismos, la invención, pero pueden ayudar al lector a comprender o implementar la invención.

- 5 Los detalles de uno o más ejemplos se exponen en los dibujos adjuntos y la descripción a continuación. Otras características, objetos y ventajas serán evidentes a partir de la descripción y los dibujos, y de las reivindicaciones.

#### Breve descripción de los dibujos

- 10 La Fig. 1 es un diagrama de bloques que ilustra un ejemplo de un sistema de codificación de video que puede utilizar las técnicas de esta divulgación.  
La Fig. 2 es un diagrama de bloques que ilustra un ejemplo de codificador de video que está configurado para implementar las técnicas de esta divulgación.  
La Fig. 3 es un diagrama de bloques que ilustra un ejemplo de un decodificador de video que está configurado para  
15 implementar las técnicas de esta divulgación.  
La Fig. 4 es un diagrama de flujo que ilustra un ejemplo de operación para generar datos de porciones para una porción de una imagen.  
La Fig. 5 es un diagrama de flujo que ilustra un ejemplo de operación para decodificar una unidad NAL de la porción codificada.  
20 La Fig. 6 es un diagrama conceptual que ilustra el procesamiento paralelo de frente de onda.  
La Fig. 7 es un diagrama conceptual que ilustra un ejemplo de orden de codificación cuando una imagen se particiona en una pluralidad de mosaicos.  
La Fig. 8 es un diagrama conceptual que ilustra un ejemplo de unidad NAL de la porción codificada.

#### 25 Descripción detallada

- Una imagen incluye una pluralidad de bloques de árbol. Los bloques de árbol están asociados con bloques de video bidimensionales dentro de la imagen. Un codificador de video divide la imagen en una pluralidad de particiones de  
30 imagen. Por ejemplo, el codificador de video puede dividir la imagen en mosaicos u ondas de procesamiento paralelo de frente de onda (WPP). En otras palabras, esta divulgación puede usar el término "partición de imagen" para referirse genéricamente a mosaicos u ondas de WPP. Las particiones de imagen están asociadas con subconjuntos no superpuestos de los bloques de árbol de la imagen. Por ejemplo, cada bloque de árbol de la imagen puede estar asociado con exactamente una de las particiones de la imagen. En las reivindicaciones independientes, la imagen se codifica mediante WPP.

- 35 El codificador de video genera una unidad de capa de abstracción de red (NAL) de porción codificada. La unidad NAL de la porción codificada puede incluir representaciones codificadas de cada bloque de árbol asociado con una porción de la imagen. Esta divulgación puede hacer referencia a una representación codificada de un bloque de árbol como un bloque de árbol codificado. Un bloque de árbol codificado puede incluir una secuencia de bits que representan el  
40 bloque de video asociado con un bloque de árbol. La secuencia de bits en un bloque de árbol codificado puede representar una secuencia de elementos de sintaxis.

- El codificador de video agrupa los bloques de árbol codificados dentro de la unidad NAL de la porción codificada en segmentos. Los segmentos están asociados con diferentes particiones de la imagen. Cada uno de los segmentos es  
45 una serie consecutiva de bits, incluidos los bits que representan una serie de uno o más bloques de árbol codificados y datos asociados. Así, la unidad NAL de la porción codificada puede incluir cada bloque de árbol codificado asociado con una primera partición de imagen seguido de cada bloque de árbol codificado asociado con una segunda partición de imagen, seguido de cada bloque de árbol codificado asociado con una tercera partición de imagen, y así sucesivamente.

- 50 De acuerdo con las técnicas de esta divulgación, el codificador de video rellena uno o más de los segmentos de modo que cada uno de los segmentos comience en un límite de byte. Cuando el codificador de video rellena un segmento, el codificador de video puede añadir bits de relleno al segmento. Los bits de relleno pueden no tener ningún significado semántico, pero pueden servir para garantizar que el siguiente segmento comience en un límite de byte. De esta  
55 forma, el codificador de video puede proporcionar alineación de bytes de mosaicos u ondas WPP cuando los mosaicos u ondas WPP se incluyen en una unidad NAL de la porción codificada para fines de procesamiento en paralelo. En las reivindicaciones independientes, la imagen se particiona en una pluralidad de ondas WPP.

- Un decodificador de video puede almacenar la unidad NAL de la porción codificada en byte de memoria direccionada.  
60 El decodificador de video puede entonces asignar dos o más de los segmentos a diferentes subprocesos de decodificación que operan en paralelo. Cada subproceso de decodificación decodifica los bloques de árbol codificados del segmento asignado al subproceso de decodificación. Debido a que cada uno de los segmentos comienza en un límite de byte, el decodificador de video puede proporcionar una dirección de memoria de un segmento a un subproceso de decodificación cuando asigna el segmento al subproceso de decodificación. De esta forma, asegurar  
65 que cada uno de los segmentos comience en un límite de byte puede permitir que el decodificador de video decodifique

los segmentos en paralelo de una manera más simple que cuando los segmentos pueden comenzar en posiciones que no están en el límite de byte.

Esto puede contrastar con los codificadores de video convencionales y los decodificadores de video convencionales que no aseguran que los segmentos comiencen en los límites de byte. Debido a que los segmentos pueden no comenzar en los límites de byte, es posible que un decodificador de video convencional que use direccionamiento de memoria por byte no pueda decodificar los bloques de árbol codificados en los segmentos en paralelo. Un decodificador de video convencional puede usar direccionamiento de memoria por bit o direccionamiento por byte más direccionamiento por bit para permitir la decodificación de los bloques de árbol codificados en los segmentos en paralelo, pero con mayores complejidades de implementación y cálculo.

Los dibujos adjuntos ilustran ejemplos. Los elementos indicados con números de referencia en los dibujos adjuntos corresponden a elementos indicados con números de referencia similares en la siguiente descripción. En esta divulgación, los elementos que tienen nombres que comienzan con palabras ordinales (por ejemplo, "primero", "segundo", "tercero", etc.) no implican necesariamente que los elementos tengan un orden particular. Más bien, tales palabras ordinales se usan simplemente para referirse a diferentes elementos del mismo tipo o similar. Además, en la siguiente descripción, la "imagen actual" puede referirse a una imagen que se está codificando o decodificando actualmente.

La Fig. 1 es un diagrama de bloques que ilustra un ejemplo de sistema 10 de codificación de video que puede utilizar las técnicas de esta divulgación. Como se describe en este documento, el término "codificador de video" se refiere genéricamente tanto a codificadores de video como a decodificadores de video. En esta divulgación, los términos "codificación de video" o "codificación" pueden referirse genéricamente a codificación de video y decodificación de video.

Como se muestra en la Fig. 1, el sistema 10 de codificación de video incluye un dispositivo 12 fuente y un dispositivo 14 de destino. El dispositivo 12 fuente genera datos de video codificados. En consecuencia, el dispositivo 12 fuente puede denominarse dispositivo de codificación de video. El dispositivo 14 de destino puede decodificar los datos de video codificados generados por el dispositivo 12 fuente. En consecuencia, el dispositivo 14 de destino puede denominarse dispositivo de decodificación de video. El dispositivo 12 fuente y el dispositivo 14 de destino pueden ser ejemplos de dispositivos de codificación de video.

El dispositivo 12 fuente y el dispositivo 14 de destino pueden comprender una amplia gama de dispositivos, que incluyen ordenadores de escritorio, dispositivos informáticos móviles, teléfonos portátiles (por ejemplo, portátiles), tabletas, decodificadores, teléfonos portátiles como los llamados teléfonos "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, ordenadores para automóviles o similares. En algunos ejemplos, el dispositivo 12 fuente y el dispositivo 14 de destino pueden estar equipados para comunicación inalámbrica.

El dispositivo 14 de destino puede recibir datos de video codificados desde el dispositivo 12 fuente a través de un canal 16. El canal 16 puede comprender un tipo de medio o dispositivo capaz de mover los datos de video codificados desde el dispositivo 12 fuente al dispositivo 14 de destino. En un ejemplo, el canal 16 puede comprender un medio de comunicación que permite que el dispositivo 12 fuente transmita datos de video codificados directamente al dispositivo 14 de destino en tiempo real. En este ejemplo, el dispositivo 12 fuente puede modular los datos de video codificados de acuerdo con un estándar de comunicación, tal como un protocolo de comunicación inalámbrica, y puede transmitir los datos de video modulados al dispositivo 14 de destino. El medio de comunicación puede comprender un medio de comunicación inalámbrico o por cable, tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión física. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área amplia o una red global como Internet. El medio de comunicación puede incluir enrutadores, conmutadores, estaciones base u otro equipo que facilite la comunicación desde el dispositivo 12 fuente al dispositivo 14 de destino.

En otro ejemplo, el canal 16 puede corresponder a un medio de almacenamiento que almacena los datos de video codificados generados por el dispositivo 12 fuente. En este ejemplo, el dispositivo 14 de destino puede acceder al medio de almacenamiento mediante acceso a disco o acceso a tarjeta. El medio de almacenamiento puede incluir una variedad de medios de almacenamiento de datos a los que se accede localmente, como discos Blu-ray, DVD, CD-ROM, memoria flash u otros medios de almacenamiento digital adecuados para almacenar datos de video codificados. En otro ejemplo, el canal 16 puede incluir un servidor de archivos u otro dispositivo de almacenamiento intermedio que almacena el video codificado generado por el dispositivo 12 fuente. En este ejemplo, el dispositivo 14 de destino puede acceder a los datos de video codificado almacenados en el servidor de archivos u otro dispositivo de almacenamiento intermedio a través de transmisión o descarga. El servidor de archivos puede ser un tipo de servidor capaz de almacenar datos de video codificados y transmitir los datos de video codificados al dispositivo 14 de destino. Los ejemplos de servidores de archivos incluyen servidores web (por ejemplo, para un sitio web), servidores de protocolo de transferencia de archivos (FTP), dispositivos de almacenamiento unidos a la red (NAS) y unidades de disco locales. El dispositivo 14 de destino puede acceder a los datos de video codificados a través de una conexión de datos estándar, incluida una conexión a Internet. Los tipos de ejemplo de conexiones de datos pueden incluir

canales inalámbricos (p. ej., conexiones Wi-Fi), conexiones por cable (p. ej., DSL, módem por cable, etc.) o combinaciones de ambas que son adecuadas para acceder a datos de video codificados almacenados en un servidor de archivos. La transmisión de datos de video codificados desde el servidor de archivos puede ser una transmisión continua, una transmisión de descarga o una combinación de ambas.

Las técnicas de esta divulgación no se limitan a aplicaciones o configuraciones inalámbricas. Las técnicas se pueden aplicar a la codificación de video en apoyo de cualquiera de una variedad de aplicaciones multimedia, tales como transmisiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de transmisión de video, por ejemplo, a través de Internet, codificación de video digital para almacenamiento en un medio de almacenamiento de datos, decodificación de video digital almacenado en un medio de almacenamiento de datos u otras aplicaciones. En algunos ejemplos, el sistema 10 de codificación de video puede configurarse para admitir transmisión de video unidireccional o bidireccional para admitir aplicaciones tales como transmisión de video, reproducción de video, radiodifusión de videos y/o videotelefonía.

En el ejemplo de la Fig. 1, el dispositivo 12 fuente incluye una fuente 18 de video, un codificador 20 de video y una interfaz 22 de salida. En algunos casos, la interfaz 22 de salida puede incluir un modulador/demodulador (módem) y/o un transmisor. En el dispositivo 12 fuente, la fuente 18 de video puede incluir una fuente tal como un dispositivo de captura de video, por ejemplo, una cámara de video, un archivo de video que contiene datos de video capturados previamente, una interfaz de alimentación de video para recibir datos de video de un proveedor de contenido de video y/ o un sistema de gráficos por ordenador para generar datos de video, o una combinación de dichas fuentes.

El codificador 20 de video puede codificar los datos de video capturados, previamente capturados o generados por ordenador. Los datos de video codificados pueden transmitirse directamente al dispositivo 14 de destino a través de la interfaz 22 de salida del dispositivo 12 fuente. Los datos de video codificados también pueden almacenarse en un medio de almacenamiento o un servidor de archivos para su posterior acceso por parte del dispositivo 14 de destino para decodificación y/o reproducción.

En el ejemplo de la Fig. 1, el dispositivo 14 de destino incluye una interfaz 28 de entrada, un decodificador 30 de video y un dispositivo 32 de visualización. En algunos casos, la interfaz 28 de entrada puede incluir un receptor y/o un módem. La interfaz 28 de entrada del dispositivo 14 de destino recibe datos de video codificados por el canal 16. Los datos de video codificados incluyen una variedad de elementos de sintaxis generados por el codificador 20 de video que representan los datos de video. Dichos elementos de sintaxis pueden incluirse con los datos de video codificados transmitidos en un medio de comunicación, almacenados en un medio de almacenamiento o almacenados en un servidor de archivos.

El dispositivo 32 de visualización puede estar integrado o puede ser externo al dispositivo 14 de destino. En algunos ejemplos, el dispositivo 14 de destino puede incluir un dispositivo de visualización integrado y también puede configurarse para interactuar con un dispositivo de visualización externo. En otros ejemplos, el dispositivo 14 de destino puede ser un dispositivo de visualización. En general, el dispositivo 32 de visualización muestra los datos de video decodificados a un usuario. El dispositivo 32 de visualización puede comprender cualquiera de una variedad de dispositivos de visualización tales como una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodo orgánico emisor de luz (OLED) u otro tipo de dispositivo de visualización.

El codificador 20 de video y el decodificador 30 de video pueden operar de acuerdo con un estándar de compresión de video, tal como el estándar de codificación de video de alta eficiencia (HEVC) actualmente en desarrollo, y pueden ajustarse a un modelo de prueba HEVC (HM). Un borrador reciente del próximo estándar HEVC, denominado "Borrador de trabajo 6 de HEVC" o "WD6", se describe en el documento JCTVC-H1003, Bross et al., "High efficiency video coding (HEVC) text specification draft 6", Joint Collaborative Team on Video Coding (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, octava reunión: San José, California, EE. UU., febrero de, 2012, que, a partir del 1 de mayo, 2012, puede descargarse a través de: [http://phenix.int-evr.fr/jct/doc\\_end\\_user/documents/8\\_San%20Jose/wg11/JCTVC-H1003-v22.zip](http://phenix.int-evr.fr/jct/doc_end_user/documents/8_San%20Jose/wg11/JCTVC-H1003-v22.zip). Alternativamente, el codificador 20 de video y el decodificador 30 de video pueden operar de acuerdo con otros estándares patentados o de la industria, como el estándar ITU-T H.264, también denominado MPEG-4, Parte 10, codificación de video avanzada (AVC) o extensiones de dichos estándares, cuando se incluyen técnicas de partición de imágenes tales como mosaicos o procesamiento paralelo de frente de onda. Las técnicas de esta divulgación, sin embargo, no se limitan a ningún estándar o técnica de codificación en particular. Otros ejemplos de estándares y técnicas de compresión de video incluyen MPEG-2, ITU-T H.263 y formatos de compresión patentados o de código abierto tales como VP8 y formatos relacionados, cuando se incluyen técnicas de partición de imágenes tales como mosaicos o procesamiento paralelo de frente de onda.

Aunque no se muestra en el ejemplo de la Fig. 1, el codificador 20 de video y el decodificador 30 de video pueden integrarse cada uno con un codificador y decodificador de audio, y pueden incluir unidades MUX-DEMUX apropiadas, u otro hardware y software, para manejar la codificación de audio y video en un flujo de datos común o flujo de datos separados. Si corresponde, en algunos ejemplos, las unidades MUX-DEMUX pueden cumplir con el protocolo multiplexor ITU H.223 u otros protocolos tales como el protocolo de datagramas de usuario (UDP).

Nuevamente, la Fig. 1 es simplemente un ejemplo y las técnicas de esta divulgación pueden aplicarse a configuraciones de codificación de video (por ejemplo, codificación de video o decodificación de video) que no necesariamente incluyen ninguna comunicación de datos entre los dispositivos de codificación y decodificación. En otros ejemplos, los datos pueden recuperarse de una memoria local, transmitirse a través de una red o similar. Un dispositivo de codificación puede codificar y almacenar datos en la memoria, y/o un dispositivo de decodificación puede recuperar y decodificar datos de la memoria. En muchos ejemplos, la codificación y decodificación la realizan dispositivos que no se comunican entre sí, sino que simplemente codifican datos en la memoria y/o recuperan y decodifican datos de la memoria.

El codificador 20 de video y el decodificador 30 de video pueden implementarse cada uno como cualquiera de una variedad de circuitos adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), matrices de puertas programables de campo (FPGA), lógica discreta, hardware o cualquier combinación de los mismos. Cuando las técnicas se implementan 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 divulgación. Cada uno del codificador 20 de video y decodificador 30 de video puede incluirse en uno o más codificadores o decodificadores, cualquiera de los cuales puede integrarse como parte de un codificador/decodificador combinado (CODEC) en un dispositivo respectivo.

Como se mencionó brevemente anteriormente, el codificador 20 de video codifica datos de video. Los datos de video pueden comprender una o más imágenes. Cada una de las imágenes es una imagen fija que forma parte de un vídeo. En algunos casos, se puede hacer referencia a una imagen como a un "cuadro" de video o un "campo" de video. Cuando el codificador 20 de video codifica los datos de video, el codificador 20 de video puede generar un flujo de bits. El flujo de bits puede incluir una secuencia de bits que forman una representación codificada de los datos de video. El flujo de bits puede incluir imágenes codificadas y datos asociados. Una imagen codificada es una representación codificada de una imagen.

Para generar el flujo de bits, el codificador 20 de video puede realizar operaciones de codificación en cada imagen en los datos de video. Cuando el codificador 20 de video realiza operaciones de codificación en las imágenes, el codificador 20 de video puede generar una serie de imágenes codificadas y datos asociados. Los datos asociados pueden incluir conjuntos de parámetros de secuencia, conjuntos de parámetros de imágenes, conjuntos de parámetros de adaptación y otras estructuras sintácticas. Un conjunto de parámetros de secuencia (SPS) puede contener parámetros aplicables a cero o más secuencias de imágenes. Las secuencias de imágenes también pueden denominarse secuencias de video codificadas, como en H.264/AVC y HEVC. Un conjunto de parámetros de imagen (PPS) puede contener parámetros aplicables a cero o más imágenes. Un conjunto de parámetros de adaptación (APS) puede contener parámetros aplicables a cero o más imágenes. Los parámetros en un APS pueden ser parámetros que tienen más probabilidades de cambiar que los parámetros en un PPS.

Para generar una imagen codificada, el codificador 20 de video puede dividir una imagen en bloques de video de igual tamaño. Un bloque de video puede ser una matriz bidimensional de muestras. Cada uno de los bloques de video está asociado con un bloque de árbol. En algunos casos, un bloque de árbol puede denominarse unidad de codificación más grande (LCU) o bloque de árbol de codificación. Los bloques de árbol de HEVC pueden ser ampliamente análogos a los macrobloques de estándares anteriores, tales como H.264/AVC. Sin embargo, un bloque de árbol no está necesariamente limitado a un tamaño particular y puede incluir una o más unidades de codificación (CU). El codificador 20 de video puede utilizar la partición de árbol cuádruple para dividir los bloques de video de los bloques de árbol en bloques de video asociados con las CU, de ahí el nombre "bloques de árbol".

En algunos ejemplos, el codificador 20 de video puede dividir una imagen en una pluralidad de porciones. Cada una de las porciones puede incluir un número entero de bloques de árbol codificados consecutivamente. En algunos casos, cada uno de las porciones puede incluir un número entero de CU codificadas consecutivamente. Como parte de la realización de una operación de codificación en una imagen, el codificador 20 de video puede realizar operaciones de codificación en cada porción de la imagen. Cuando el codificador 20 de video realiza una operación de codificación en una porción, el codificador 20 de video puede generar datos codificados asociados con la porción. Los datos codificados asociados con la porción pueden denominarse como una "porción codificada".

Para generar una la porción codificada, el codificador 20 de video puede realizar operaciones de codificación en cada bloque de árbol en una porción. Cuando el codificador 20 de video realiza una operación de codificación en un bloque de árbol, el codificador 20 de video puede generar un bloque de árbol codificado. El bloque de árbol codificado puede comprender datos que representan una versión codificada del bloque de árbol.

Cuando el codificador 20 de video genera una porción codificada, el codificador 20 de video puede realizar operaciones de codificación en (es decir, codificar) los bloques de árbol en la porción de acuerdo con un orden de escaneo de trama. En otras palabras, el codificador 20 de video puede codificar los bloques de árbol de la porción en un orden que avanza de izquierda a derecha a lo largo de una fila superior de bloques de árbol en la porción, luego continúa de izquierda a derecha a lo largo de la siguiente fila inferior de bloques de árbol, y así sucesivamente hasta que el

codificador 20 de video haya codificado cada uno de los bloques de árbol en la porción. El uso de un escaneo de trama para codificar los bloques de árbol no está incluido en la invención reivindicada.

Como resultado de la codificación de los bloques de árbol de acuerdo con el orden de escaneo de trama, los bloques de árbol situados encima y a la izquierda de un bloque de árbol dado pueden haberse codificado, pero los bloques de árbol situados debajo y a la derecha de dicho bloque de árbol todavía no se han codificado. En consecuencia, el codificador 20 de video puede acceder a la información generada mediante la codificación de bloques de árbol por encima y a la izquierda del bloque de árbol dado al codificar el bloque de árbol dado. Sin embargo, es posible que el codificador 20 de video no pueda acceder a la información generada mediante la codificación de bloques de árbol debajo y a la derecha del bloque de árbol dado al codificar el bloque de árbol dado.

Para generar un bloque de árbol codificado, el codificador 20 de video puede realizar recursivamente la partición de árbol cuádruple en el bloque de video del bloque de árbol para dividir el bloque de video en bloques de video progresivamente más pequeños. Cada uno de los bloques de video más pequeños puede estar asociado con una CU diferente. Por ejemplo, el codificador 20 de video puede dividir el bloque de video de un bloque de árbol en cuatro subbloques de igual tamaño, dividir uno o más de los subbloques en cuatro subsubbloques de igual tamaño, y así sucesivamente. Una CU dividida puede ser una CU cuyo bloque de video está dividido en bloques de video asociados con otras CU. Una CU no dividida puede ser una CU cuyo bloque de video no está dividido en bloques de video asociados con otras CU.

Uno o más elementos de sintaxis en el flujo de bits pueden indicar un número máximo de veces que el codificador 20 de video puede dividir el bloque de video de un bloque de árbol. Un bloque de video de una CU puede tener forma cuadrada. El tamaño del bloque de video de una CU (es decir, el tamaño de la CU) puede variar desde 8x8 píxeles hasta el tamaño de un bloque de video de un bloque de árbol (es decir, el tamaño del bloque de árbol) con un máximo de 64x64 píxeles o mayor.

El codificador 20 de video puede realizar operaciones de codificación en (es decir, codificar) cada CU de un bloque de árbol de acuerdo con un orden de escaneo z. En otras palabras, el codificador 20 de video puede codificar una CU superior izquierda, una CU superior derecha, una CU inferior izquierda y luego una CU inferior derecha, en ese orden. Cuando el codificador 20 de video realiza una operación de codificación en una CU dividida, el codificador 20 de video puede codificar CU asociadas con subbloques del bloque de video de la CU dividida de acuerdo con el orden de escaneo z. En otras palabras, el codificador 20 de video puede codificar una CU asociada con un subbloque superior izquierdo, una CU asociada con un subbloque superior derecho, una CU asociada con un subbloque inferior izquierdo y luego una CU asociada con un subbloque inferior derecho, en ese orden.

Como resultado de codificar las CU de un bloque de árbol de acuerdo con un orden de escaneo z, las CU arriba, arriba y a la izquierda, arriba y a la derecha, izquierda y abajo y a la izquierda de una CU dada puede haber sido codificada. Las CU debajo y a la derecha de la CU dada aún no se han codificado. En consecuencia, el codificador 20 de video puede acceder a la información generada al codificar algunas CU que son vecinas de la CU dada al codificar la CU dada. Sin embargo, el codificador 20 de video puede ser incapaz de acceder a la información generada al codificar otras CU cercanas a la CU dada cuando codifica la CU dada.

Cuando el codificador 20 de video codifica una CU no dividida, el codificador 20 de video puede generar una o más unidades de predicción (PU) para la CU. Cada una de las PU de la CU puede estar asociada con un bloque de video diferente dentro del bloque de video de la CU. El codificador 20 de video puede generar un bloque de video predicho para cada PU de la CU. El bloque de video predicho de una PU puede ser un bloque de muestras. El codificador 20 de video puede usar predicción dentro o predicción entre para generar el bloque de video predicho para una PU.

Cuando el codificador 20 de video usa predicción dentro para generar el bloque de video predicho de una PU, el codificador 20 de video puede generar el bloque de video predicho de la PU basándose en muestras decodificadas de la imagen asociada con la PU. Si el codificador 20 de video usa predicción dentro para generar bloques de video predichos de las PU de una CU, la CU es una CU predicha internamente. Cuando el codificador 20 de video usa la interpredicción para generar el bloque de video predicho de la PU, el codificador 20 de video puede generar el bloque de video predicho de la PU basándose en muestras decodificadas de una o más imágenes distintas de la imagen asociada con la PU. Si el codificador 20 de video usa interpredicción para generar bloques de video predichos de las PU de una CU, la CU es una CU interpredicha.

Además, cuando el codificador 20 de video usa la interpredicción para generar un bloque de video predicho para una PU, el codificador 20 de video puede generar información de movimiento para la PU. La información de movimiento para una PU puede indicar uno o más bloques de referencia de la PU. Cada bloque de referencia de la PU puede ser un bloque de video dentro de una imagen de referencia. La imagen de referencia puede ser una imagen distinta de la imagen asociada con la PU. En algunos casos, un bloque de referencia de una PU también puede denominarse "muestra de referencia" de la PU. El codificador 20 de video puede generar el bloque de video predicho para la PU basándose en los bloques de referencia de la PU.

Después de que el codificador 20 de video genera bloques de video predichos para una o más PU de una CU, el codificador 20 de video puede generar datos residuales para la CU basándose en los bloques de video predichos para las PU de la CU. Los datos residuales para la CU pueden indicar diferencias entre muestras en los bloques de video predichos para las PU de la CU y el bloque de video original de la CU.

Además, como parte de la realización de una operación de codificación en una CU no dividida, el codificador 20 de video puede realizar una partición de árbol cuádruple recursiva en los datos residuales de la CU para dividir los datos residuales de la CU en uno o más bloques de datos residuales (es decir, bloques de video residuales) asociados con unidades de transformación (TU) de la CU. Cada TU de una CU puede estar asociada con un bloque de video residual diferente.

El codificador 20 de video puede aplicar una o más transformaciones a bloques de video residuales asociados con las TU para generar bloques de coeficientes de transformación (es decir, bloques de coeficientes de transformación) asociados con las TU. Conceptualmente, un bloque del coeficiente de transformación puede ser una matriz bidimensional (2D) de coeficientes de transformación.

Después de generar un bloque del coeficiente de transformación, el codificador 20 de video puede realizar un proceso de cuantificación en el bloque del coeficiente de transformación. La cuantificación generalmente se refiere a un proceso en el que los coeficientes de transformación se cuantifican para reducir posiblemente la cantidad de datos utilizados para representar los coeficientes de transformación, proporcionando una mayor compresión. El proceso de cuantificación puede reducir la profundidad de bits asociada con algunos o todos los coeficientes de transformación. Por ejemplo, un coeficiente de transformación de  $n$  bits puede redondearse hacia abajo a un coeficiente de transformación de  $m$  bits durante la cuantificación, donde  $n$  es mayor que  $m$ .

El codificador 20 de video puede asociar cada CU con un valor de parámetro de cuantificación (QP). El valor de QP asociado con una CU puede determinar cómo el codificador 20 de video cuantifica los bloques de coeficientes de transformación asociados con la CU. El codificador 20 de video puede ajustar el grado de cuantificación aplicado a los bloques de coeficientes de transformación asociados con una CU ajustando el valor de QP asociado con la CU.

Después de que el codificador 20 de video cuantifica un bloque del coeficiente de transformación, el codificador 20 de video puede generar conjuntos de elementos de sintaxis que representan los coeficientes de transformación en el bloque del coeficiente de transformación cuantificados. El codificador 20 de video puede aplicar operaciones de codificación de entropía, tales como operaciones de codificación aritmética binaria adaptable al contexto (CABAC), a algunos de estos elementos de sintaxis.

El flujo de bits generado por el codificador 20 de video puede incluir una serie de unidades de capa de abstracción de red (NAL). Cada una de las unidades NAL puede ser una estructura sintáctica que contenga una indicación de un tipo de datos en la unidad NAL y bytes que contengan los datos. Por ejemplo, una unidad NAL puede contener datos que representan un conjunto de parámetros de secuencia, un conjunto de parámetros de imagen, una porción codificada, uno o más mensajes de información de mejora suplementaria (SEI), un delimitador de unidad de acceso, datos de relleno u otro tipo de datos. Los datos en una unidad NAL pueden incluir varias estructuras sintácticas.

El decodificador 30 de video puede recibir el flujo de bits generado por el codificador 20 de video. El flujo de bits puede incluir una representación codificada de los datos de video codificados por el codificador 20 de video. Cuando el decodificador 30 de video recibe el flujo de bits, el decodificador 30 de video puede realizar una operación de análisis en el flujo de bits. Cuando el decodificador 30 de video realiza la operación de análisis, el decodificador 30 de video puede extraer elementos de sintaxis del flujo de bits. El decodificador 30 de video puede reconstruir las imágenes de los datos de video basándose en los elementos de sintaxis extraídos del flujo de bits. El proceso para reconstruir los datos de video en base a los elementos de sintaxis puede ser generalmente recíproco al proceso realizado por el codificador 20 de video para generar los elementos de sintaxis.

Después de que el decodificador 30 de video extrae los elementos de sintaxis asociados con una CU, el decodificador 30 de video puede generar bloques de video predichos para las PU de la CU basándose en los elementos de sintaxis. Además, el decodificador 30 de video puede cuantificar inversamente los bloques de coeficientes de transformación asociados con las TU de la CU. El decodificador 30 de video puede realizar transformaciones inversas en los bloques de coeficientes de transformación para reconstruir bloques de video residuales asociados con las TU de la CU. Después de generar los bloques de video predichos y reconstruir los bloques de video residuales, el decodificador 30 de video puede reconstruir el bloque de video de la CU basándose en los bloques de video predichos y los bloques de video residuales. De esta forma, el decodificador 30 de video puede reconstruir los bloques de video de las CU basándose en los elementos de sintaxis en el flujo de bits.

El codificador 20 de video puede dividir la imagen actual en una pluralidad de particiones de imagen. Las particiones de imagen pueden asociarse con subconjuntos no superpuestos de los bloques de árbol de la imagen actual. El codificador 20 de video puede dividir la imagen actual en una pluralidad de particiones de imagen de varias maneras. Como se describe a continuación, el codificador 20 de video puede dividir la imagen actual en una pluralidad de mosaicos o en una pluralidad de ondas de procesamiento paralelo de frente de onda (WPP). Esta divulgación puede



usar el término "partición de imagen" para referirse genéricamente tanto a mosaicos como a ondas WPP. El proceso de dividir la imagen actual en particiones de imagen puede denominarse "partición" de la imagen actual en particiones de imagen.

- 5 Como se mencionó anteriormente, el codificador 20 de video puede dividir la imagen actual en uno o más mosaicos. Cada uno de los mosaicos puede comprender un número entero de bloques de árbol en la imagen actual. El codificador 20 de vídeo puede dividir la imagen actual en mosaicos definiendo dos o más límites de mosaicos verticales y dos o más límites de mosaicos horizontales. Cada lado vertical de la imagen actual puede considerarse como un límite de mosaico vertical. Cada lado horizontal de la imagen actual puede considerarse como un límite de mosaico horizontal. Por ejemplo, si el codificador 20 de vídeo define cuatro límites de mosaicos verticales y tres límites de mosaicos horizontales para la imagen actual, la imagen actual se divide en seis mosaicos.

15 Un codificador de video, tal como un codificador 20 de video o un decodificador 30 de video, puede codificar los mosaicos de la imagen actual de acuerdo con el orden de escaneo de la trama. Además, cuando el codificador de video codifica un mosaico, el codificador de video puede codificar cada bloque de árbol dentro del mosaico de acuerdo con un orden de escaneo de trama. De esta forma, el codificador de vídeo puede codificar cada bloque de árbol de un mosaico dado de la imagen actual antes de codificar cualquier bloque de árbol de otro mosaico de la imagen actual. En consecuencia, el orden en el que el codificador de video codifica los bloques de árbol de la imagen actual puede ser diferente cuando el codificador de video particiona la imagen actual en múltiples mosaicos que cuando el codificador de video no particiona la imagen actual en múltiples mosaicos.

25 Además, en algunos casos, el codificador de video puede usar información asociada con las CU espacialmente vecinas para realizar una intrapredicción en una CU dada en la imagen actual, siempre que la CU dada y las CU espacialmente vecinas pertenezcan al mismo mosaico. Las CU espacialmente vecinas son CU que pertenecen a la porción actual de la imagen actual. En algunos casos, el codificador de video puede usar información asociada con CU espacialmente vecinas para seleccionar un contexto para la codificación CABAC de un elemento de sintaxis de la CU dada, siempre que la CU dada y las CU espacialmente vecinas estén dentro del mismo mosaico. Debido a estas restricciones, el codificador de video puede ser capaz de codificar en bloques de árbol paralelos de múltiples mosaicos.

30 De acuerdo con la invención, el codificador de vídeo puede codificar la imagen actual usando procesamiento paralelo de frente de onda (WPP). Cuando el codificador de video codifica la imagen actual usando WPP, el codificador de video puede dividir los bloques de árbol de la imagen actual en una pluralidad de "ondas WPP". Cada una de las ondas WPP puede corresponder a una fila diferente de bloques de árbol en la imagen actual. Cuando el codificador de video codifica la imagen actual utilizando WPP, el codificador de video puede comenzar a codificar una fila superior de bloques de árbol. Cuando el codificador de video ha codificado dos o más bloques de árbol de la fila superior, el codificador de video puede comenzar a codificar una segunda fila de bloques de árbol en paralelo con la codificación de la fila superior de bloques de árbol. Cuando el codificador de video ha codificado dos o más bloques de árbol de la fila segunda a la superior, el codificador de video puede comenzar a codificar una fila de bloque de árbol tercera a la superior en paralelo con la codificación de las filas superiores de bloques de árbol. Este patrón puede continuar hacia abajo en las filas de bloques de árbol en la imagen actual.

45 Cuando el codificador de video está codificando la imagen actual usando WPP, el codificador de video puede usar información asociada con las CU espacialmente vecinas fuera de un bloque de árbol actual para realizar una intrapredicción en una CU dada en el bloque de árbol actual, siempre que las CU espacialmente vecinas estén a la izquierda, arriba a la izquierda, arriba o arriba a la derecha del bloque de árbol actual. Si el bloque de árbol actual es el bloque de árbol más a la izquierda en una fila que no sea la fila más superior, el codificador de video puede usar información asociada con el segundo bloque de árbol de la fila inmediatamente superior para seleccionar un contexto para que CABAC codifique un elemento de sintaxis del bloque de árbol actual. De lo contrario, si el bloque de árbol actual no es el bloque de árbol más a la izquierda en la fila, el codificador de video puede usar información asociada con un bloque de árbol a la izquierda del bloque de árbol actual para seleccionar un contexto para que CABAC codifique un elemento de sintaxis del bloque de árbol actual. De esta forma, el codificador de video puede inicializar los estados de CABAC de una fila basándose en los estados de CABAC de la fila inmediatamente superior después de codificar dos o más bloques de árbol de la fila inmediatamente superior.

55 En algunos ejemplos, cuando el codificador de video está codificando la imagen actual usando WPP, los únicos límites de mosaico de la imagen actual son los bordes horizontales y verticales de la imagen actual. Por lo tanto, el único mosaico de la imagen actual puede tener el mismo tamaño que la imagen actual. El codificador de vídeo puede dividir la imagen actual y, por lo tanto, el mosaico único de la imagen actual, en múltiples ondas WPP.

60 Como se mencionó anteriormente, el codificador 20 de video puede generar una unidad NAL de la porción codificada que incluye una representación codificada de una porción. La porción puede estar asociada con un número entero de bloques de árbol codificados consecutivamente. La unidad NAL de la porción codificada puede incluir un encabezado de la porción y datos de la porción. Los datos de la porción pueden incluir representaciones codificadas de cada bloque de árbol asociado con la porción. El codificador 20 de video puede generar la unidad NAL de la porción codificada en la que dichas representaciones codificadas de los bloques de árbol se agrupan dentro de los datos de la porción en segmentos de acuerdo con las particiones de imagen a las que pertenecen los bloques de árbol. Por ejemplo, la unidad

NAL de la porción codificada puede incluir cada bloque de árbol codificado asociado con una primera partición de imagen seguido de cada bloque de árbol codificado asociado con una segunda partición de imagen, seguido de cada bloque de árbol codificado asociado con una tercera partición de imagen, y así sucesivamente.

De acuerdo con la invención, el codificador 20 de video rellena uno o más de los segmentos de manera que cada uno de los segmentos comienza en un límite de byte. La unidad NAL de la porción codificada se divide en una serie de bytes. Un segmento comienza en un límite de byte cuando un primer bit del segmento es el primer bit de uno de los bytes de la unidad NAL de la porción codificada. Además, un segmento está alineado en bytes si el primer bit de un segmento es el primer bit de uno de los bytes de la unidad NAL de la porción codificada. Cuando el codificador 20 de video rellena un segmento, el codificador 20 de video puede añadir bits de relleno al segmento. Por ejemplo, el codificador 20 de video puede agregar uno o más bits de relleno a un segmento de manera que el número de bits en el segmento sea divisible por ocho sin dejar un resto. Los bits de relleno pueden no tener ningún significado semántico, pero pueden servir para garantizar que el siguiente segmento comience en un límite de byte.

Cuando el decodificador 30 de video recibe la unidad NAL de la porción codificada, el decodificador 30 de video puede almacenar la unidad NAL de la porción codificada en la memoria. Para decodificar las particiones de imagen en paralelo, el decodificador 30 de video puede asignar los segmentos a diferentes subprocesos de decodificación que corren en paralelo. Para asignar los segmentos a diferentes subprocesos de decodificación, el decodificador 30 de video puede necesitar indicar direcciones de memoria asociadas con los comienzos de los segmentos. El decodificador 30 de video puede usar direccionamiento de memoria por bytes. En consecuencia, el decodificador 30 de video puede no ser capaz de indicar la dirección de memoria asociada con el comienzo de un segmento si el comienzo del segmento ocurre dentro de un byte. Por lo tanto, el decodificador 30 de video puede no ser capaz de decodificar los bloques de árbol codificados en los segmentos en paralelo si uno o más de los segmentos comienzan dentro de un byte. Alternativamente, el decodificador 30 de video puede usar direccionamiento de memoria por bits o direccionamiento por bytes más direccionamiento por bits para permitir la decodificación de los bloques de árbol codificados en los segmentos en paralelo, pero con mayores complejidades de implementación y cálculo.

De esta manera, el codificador 20 de video divide una imagen en una pluralidad de particiones de imagen. En las reivindicaciones independientes, el codificador de video particiona una imagen en una pluralidad de ondas WPP. La imagen tiene una pluralidad de bloques de árbol. Las particiones de imagen están asociadas con subconjuntos no superpuestos de los bloques de árbol de la imagen. El codificador 20 de video genera una unidad NAL de la porción codificada que incluye representaciones codificadas de los bloques de árbol que están asociados con una porción de la imagen. Las representaciones codificadas de los bloques de árbol se agrupan dentro de la unidad NAL de la porción codificada en segmentos asociados con diferentes particiones de imagen. Uno o más de los segmentos se rellenan de modo que cada uno de los segmentos comience en un límite de byte.

Además, el decodificador 30 de video almacena una unidad NAL de la porción codificada que incluye representaciones codificadas de bloques de árbol asociados con una porción de una imagen. La imagen se divide en una pluralidad de particiones de imagen. En las reivindicaciones independientes, la imagen se particiona en una pluralidad de ondas WPP. Las representaciones codificadas de los bloques de árbol se agrupan en segmentos asociados con diferentes particiones de imagen. Uno o más de los segmentos se rellenan de manera que cada uno de los segmentos comience en un límite de byte. El decodificador 30 de video decodifica las representaciones codificadas de los bloques de árbol. En algunos casos, el decodificador 30 de video puede decodificar las representaciones codificadas de los bloques de árbol en dos o más de los segmentos en paralelo.

La Fig. 2 es un diagrama de bloques que ilustra un ejemplo de un codificador 20 de video que está configurado para implementar las técnicas de esta divulgación. La Fig. 2 se proporciona con fines explicativos y no debe considerarse como una limitación de las técnicas tal como se ejemplifican y describen ampliamente en esta divulgación. A efectos de explicación, esta divulgación describe el codificador 20 de video en el contexto de la codificación HEVC. Sin embargo, las técnicas de esta divulgación pueden ser aplicables a otros estándares o métodos de codificación.

En el ejemplo de la Fig. 2, el codificador 20 de video incluye una pluralidad de componentes funcionales. Los componentes funcionales del codificador 20 de video incluyen un módulo 100 de predicción, un módulo 102 de generación residual, un módulo 104 de transformación, un módulo 106 de cuantificación, un módulo 108 de cuantificación inversa, un módulo 110 de transformación inversa, un módulo 112 de reconstrucción, un módulo 113 de filtro, un búfer 114 de imagen decodificada y un módulo 116 de codificación de entropía. El módulo 100 de predicción incluye un módulo 121 de interpretación, un módulo 122 de estimación de movimiento, un módulo 124 de compensación de movimiento y un módulo 126 de intrapredicción. En otros ejemplos, el codificador 20 de video puede incluir más, menos o diferentes componentes funcionales. Además, el módulo 122 de estimación de movimiento y el módulo 124 de compensación de movimiento pueden estar altamente integrados, pero están representados en el ejemplo de la Fig. 2 por separado para propósitos de explicación.

El codificador 20 de video puede recibir datos de video. El codificador 20 de video puede recibir los datos de video de varias fuentes. Por ejemplo, el codificador 20 de video puede recibir los datos de video de la fuente 18 de video (Fig. 1) u otra fuente. Los datos de video pueden representar una serie de imágenes. Para codificar los datos de video, el codificador 20 de video puede realizar una operación de codificación en cada una de las imágenes. Como parte de la

realización de la operación de codificación en una imagen, el codificador 20 de vídeo puede realizar operaciones de codificación en cada porción de la imagen. Como parte de la realización de una operación de codificación en una porción, el codificador 20 de vídeo puede realizar operaciones de codificación en bloques de árbol en la porción.

5 Como parte de la realización de una operación de codificación en un bloque de árbol, el módulo 100 de predicción puede realizar una partición de árbol cuádruple en el bloque de vídeo del bloque de árbol para dividir el bloque de vídeo en bloques de vídeo progresivamente más pequeños. Cada uno de los bloques de vídeo más pequeños puede estar asociado con una CU diferente. Por ejemplo, el módulo 100 de predicción puede particionar un bloque de vídeo de un bloque de árbol en cuatro subbloques de igual tamaño, particionar uno o más de los subbloques en cuatro subsubbloques de igual tamaño, y así sucesivamente.

10 Los tamaños de los bloques de vídeo asociados con las CU pueden variar desde 8x8 muestras hasta el tamaño del bloque de árbol con un máximo de 64x64 muestras o más. En esta divulgación, "NxN" y "N por N" se pueden usar indistintamente para referirse a las dimensiones de muestra de un bloque de vídeo en términos de dimensiones verticales y horizontales, por ejemplo, 16x16 muestras o 16 por 16 muestras. En general, un bloque de vídeo de 16x16 tiene dieciséis muestras en dirección vertical ( $y = 16$ ) y dieciséis muestras en dirección horizontal ( $x = 16$ ). Del mismo modo, un bloque NxN generalmente tiene N muestras en dirección vertical y N muestras en dirección horizontal, donde N representa un valor entero no negativo.

15 Además, como parte de la realización de la operación de codificación en un bloque de árbol, el módulo 100 de predicción puede generar una estructura de datos de árbol cuádruple jerárquica para el bloque de árbol. Por ejemplo, un bloque de árbol puede corresponder a un nodo raíz de la estructura de datos de árbol cuádruple. Si el módulo 100 de predicción particiona el bloque de vídeo del bloque de árbol en cuatro subbloques, el nodo raíz tiene cuatro nodos secundarios en la estructura de datos de árbol cuádruple. Cada uno de los nodos secundarios corresponde a una CU asociada a uno de los subbloques. Si el módulo 100 de predicción particiona uno de los subbloques en cuatro subsubbloques, el nodo correspondiente a la CU asociada al subbloque puede tener cuatro nodos secundarios, cada uno de los cuales corresponde a una CU asociada a uno de los subsubbloques.

20 Cada nodo de la estructura de datos de árbol cuádruple puede contener datos de sintaxis (por ejemplo, elementos de sintaxis) para el bloque de árbol o CU correspondiente. Por ejemplo, un nodo en el árbol cuádruple puede incluir una bandera de división que indica si el bloque de vídeo de la CU correspondiente al nodo está particionado (es decir, dividido) en cuatro subbloques. Los elementos de sintaxis para una CU pueden definirse recursivamente y pueden depender de si el bloque de vídeo de la CU está dividido en subbloques. Una CU cuyo bloque de vídeo no está particionado puede corresponder a un nodo hoja en la estructura de datos árbol cuádruple. Un bloque de árbol codificado puede incluir datos basados en la estructura de datos de árbol cuádruple para un bloque de árbol correspondiente.

25 El codificador 20 de vídeo puede realizar operaciones de codificación en cada CU no particionada de un bloque de árbol. Cuando el codificador 20 de vídeo realiza una operación de codificación en una CU no particionada, el codificador 20 de vídeo genera datos que representan una representación codificada de la CU no particionada.

30 Como parte de la realización de una operación de codificación en una CU, el módulo 100 de predicción puede particionar el bloque de vídeo de la CU entre una o más PU de la CU. El codificador 20 de vídeo y el decodificador 30 de vídeo pueden soportar varios tamaños de PU. Suponiendo que el tamaño de una CU particular es  $2N \times 2N$ , el codificador 20 de vídeo y el decodificador 30 de vídeo pueden soportar tamaños de PU de  $2N \times 2N$  o  $N \times N$ , y la interpredicción en tamaños de PU simétricos de  $2N \times 2N$ ,  $2N \times N$ ,  $N \times 2N$ ,  $N \times N$ ,  $2N \times nU$ ,  $nL \times 2N$ ,  $nR \times 2N$ , o similar. El codificador 20 de vídeo y el decodificador 30 de vídeo también pueden soportar la partición asimétrica para tamaños de PU de  $2N \times nU$ ,  $2N \times nD$ ,  $nL \times 2N$  y  $nR \times 2N$ . En algunos ejemplos, el módulo 100 de predicción puede realizar una partición geométrica para particionar el bloque de vídeo de una CU entre las PU de la CU a lo largo de un límite que no se encuentra con los lados del bloque de vídeo de la CU en ángulos rectos.

35 El módulo 121 de interpredicción puede realizar interpredicción en cada PU de la CU. La interpredicción puede proporcionar compresión temporal. Para realizar una interpredicción en una PU, el módulo 122 de estimación de movimiento puede generar información de movimiento para la PU. El módulo 124 de compensación de movimiento puede generar un bloque de vídeo predicho para la PU basado en la información de movimiento y muestras decodificadas de imágenes distintas a la imagen asociada con la CU (es decir, imágenes de referencia). En esta divulgación, un bloque de vídeo predicho generado por el módulo 124 de compensación de movimiento puede denominarse bloque de vídeo interpredicho.

40 Las porciones pueden ser porciones I, porciones P o porciones B. El módulo 122 de estimación de movimiento y el módulo 124 de compensación de movimiento pueden realizar diferentes operaciones para una PU de una CU dependiendo de si la PU está en una porción I, una porción P o una porción B. En una porción I, todas las PU son intrapredichas. Por lo tanto, si la PU está en una porción I, el módulo 122 de estimación de movimiento y el módulo 124 de compensación de movimiento no realizan interpredicción en la PU.

Si la PU está en una porción P, la imagen que contiene la PU se asocia con una lista de imágenes de referencia denominada "lista 0". Cada una de las imágenes de referencia en la lista 0 contiene muestras que pueden usarse para la interpolación de otras imágenes. Cuando el módulo 122 de estimación de movimiento realiza la operación de estimación de movimiento con respecto a una PU en una porción P, el módulo 122 de estimación de movimiento puede buscar en las imágenes de referencia en la lista 0 un bloque de referencia para la PU. El bloque de referencia de la PU puede ser un conjunto de muestras, por ejemplo, un bloque de muestras, que más se corresponda con las muestras en el bloque de video de la PU. El módulo 122 de estimación de movimiento puede usar una variedad de métricas para determinar qué tan cerca se corresponde un conjunto de muestras en una imagen de referencia con las muestras en el bloque de video de una PU. Por ejemplo, el módulo 122 de estimación de movimiento puede determinar qué tan cerca corresponde un conjunto de muestras en una imagen de referencia a las muestras en el bloque de video de una PU por suma de diferencia absoluta (SAD), suma de diferencia cuadrada (SSD) u otras métricas de diferencia.

Después de identificar un bloque de referencia de una PU en una porción P, el módulo 122 de estimación de movimiento puede generar un índice de referencia que indica la imagen de referencia en la lista 0 que contiene el bloque de referencia y un vector de movimiento que indica un desplazamiento espacial entre la PU y el bloque de referencia. En varios ejemplos, el módulo 122 de estimación de movimiento puede generar vectores de movimiento con diversos grados de precisión. Por ejemplo, el módulo 122 de estimación de movimiento puede generar vectores de movimiento con una precisión de muestra de un cuarto, una precisión de muestra de un octavo u otra precisión de muestra fraccionaria. En el caso de precisión de muestra fraccionaria, los valores del bloque de referencia pueden interpolarse a partir de valores de muestra de posición entera en la imagen de referencia. El módulo 122 de estimación de movimiento puede emitir el índice de referencia y el vector de movimiento como la información de movimiento de la PU. El módulo 124 de compensación de movimiento puede generar un bloque de video predicho de la PU con base en el bloque de referencia identificado por la información de movimiento de la PU.

Si la PU está en una porción B, la imagen que contiene la PU puede estar asociada con dos listas de imágenes de referencia, denominadas "lista 0" y "lista 1". En algunos ejemplos, una imagen que contiene una porción B se puede asociar con una lista de combinación que es una combinación de la lista 0 y la lista 1.

Además, si la PU está en una porción B, el módulo 122 de estimación de movimiento puede realizar una predicción unidireccional o una predicción bidireccional para la PU. Cuando el módulo 122 de estimación de movimiento realiza una predicción unidireccional para la PU, el módulo 122 de estimación de movimiento puede buscar en las imágenes de referencia de la lista 0 o la lista 1 un bloque de referencia para la PU. El módulo 122 de estimación de movimiento puede entonces generar un índice de referencia que indica la imagen de referencia en la lista 0 o la lista 1 que contiene el bloque de referencia y un vector de movimiento que indica un desplazamiento espacial entre la PU y el bloque de referencia. El módulo 122 de estimación de movimiento puede generar el índice de referencia, un indicador de dirección de predicción y el vector de movimiento como la información de movimiento de la PU. El indicador de dirección de predicción puede indicar si el índice de referencia indica una imagen de referencia en la lista 0 o en la lista 1. El módulo 124 de compensación de movimiento puede generar el bloque de video predicho de la PU basándose en el bloque de referencia indicado por la información de movimiento de la PU.

Cuando el módulo 122 de estimación de movimiento realiza una predicción bidireccional para una PU, el módulo 122 de estimación de movimiento puede buscar en las imágenes de referencia en la lista 0 un bloque de referencia para la PU y también puede buscar en las imágenes de referencia en la lista 1 otro bloque de referencia para la PU. El módulo 122 de estimación de movimiento puede entonces generar índices de referencia que indican las imágenes de referencia en la lista 0 y la lista 1 que contienen los bloques de referencia y vectores de movimiento que indican desplazamientos espaciales entre los bloques de referencia y la PU. El módulo 122 de estimación de movimiento puede emitir los índices de referencia y los vectores de movimiento de la PU como información de movimiento de la PU. El módulo 124 de compensación de movimiento puede generar el bloque de video predicho de la PU basándose en los bloques de referencia indicados por la información de movimiento de la PU.

En algunos casos, el módulo 122 de estimación de movimiento no emite un conjunto completo de información de movimiento para una PU al módulo 116 de codificación de entropía. Más bien, el módulo 122 de estimación de movimiento puede señalar la información de movimiento de una PU con referencia a la información de movimiento de otra PU. Por ejemplo, el módulo 122 de estimación de movimiento puede determinar que la información de movimiento de la PU es suficientemente similar a la información de movimiento de una PU vecina. En este ejemplo, el módulo 122 de estimación de movimiento puede indicar, en una estructura de sintaxis asociada con la PU, un valor que indica al decodificador 30 de video que la PU tiene la misma información de movimiento que la PU vecina. En otro ejemplo, el módulo 122 de estimación de movimiento puede identificar, en una estructura de sintaxis asociada con la PU, una PU vecina y una diferencia de vector de movimiento (MVD). La diferencia de vector de movimiento indica una diferencia entre el vector de movimiento de la PU y el vector de movimiento de la PU vecina indicada. El decodificador 30 de video puede usar el vector de movimiento de la PU vecina indicada y la diferencia de vector de movimiento para determinar el vector de movimiento de la PU. Haciendo referencia a la información de movimiento de una primera PU cuando se señala la información de movimiento de una segunda PU, el codificador 20 de video puede señalar la información de movimiento de la segunda PU utilizando menos bits.

Como parte de la realización de una operación de codificación en una CU, el módulo 126 de intrapredicción puede realizar intrapredicción en las PU de la CU. La intrapredicción puede proporcionar compresión espacial. Cuando el módulo 126 de intrapredicción realiza intrapredicción en una PU, el módulo 126 de intrapredicción puede generar datos de predicción para la PU basándose en muestras decodificadas de otras PU en la misma imagen. Los datos de predicción para la PU pueden incluir un bloque de video predicho y varios elementos de sintaxis. El módulo 126 de intrapredicción puede realizar intrapredicción en las PU en las porciones I, porciones P y porciones B.

Para realizar la intrapredicción en una PU, el módulo 126 de intrapredicción puede usar múltiples modos de intrapredicción para generar múltiples conjuntos de datos de predicción para la PU. Cuando el módulo 126 de intrapredicción usa un modo de intrapredicción para generar un conjunto de datos de predicción para la PU, el módulo 126 de intrapredicción puede extender muestras de bloques de video de PU vecinas a través del bloque de video de la PU en una dirección y/o gradiente asociado con el modo de intrapredicción. Las PU vecinas pueden estar arriba, arriba y a la derecha, arriba y a la izquierda o a la izquierda de la PU, suponiendo un orden de codificación de izquierda a derecha y de arriba a abajo para PU, CU y bloques de árbol. El módulo 126 de intrapredicción puede usar varios números de modos de intrapredicción, por ejemplo, 33 modos de intrapredicción direccional, dependiendo del tamaño de la PU.

El módulo 100 de predicción puede seleccionar los datos de predicción para una PU entre los datos de predicción generados por el módulo 124 de compensación de movimiento para la PU o los datos de predicción generados por el módulo 126 de intrapredicción para la PU. En algunos ejemplos, el módulo 100 de predicción selecciona los datos de predicción para la PU basándose en las métricas de tasa/distorsión de los conjuntos de datos de predicción.

Si el módulo 100 de predicción selecciona datos de predicción generados por el módulo 126 de intrapredicción, el módulo 100 de predicción puede señalar el modo de intrapredicción que se usó para generar los datos de predicción para las PU, es decir, el modo de intrapredicción seleccionado. El módulo 100 de predicción puede señalar el modo de intrapredicción seleccionado de varias formas. Por ejemplo, es probable que el modo de intrapredicción seleccionado sea el mismo que el modo de intrapredicción de una PU vecina. En otras palabras, el modo de intrapredicción de la PU vecina puede ser el modo más probable para la PU actual. Por lo tanto, el módulo 100 de predicción puede generar un elemento de sintaxis para indicar que el modo de intrapredicción seleccionado es el mismo que el modo de intrapredicción de la PU vecina.

Después de que el módulo 100 de predicción selecciona los datos de predicción para las PU de una CU, el módulo 102 de generación residual puede generar datos residuales para la CU restando los bloques de video predichos de las PU de la CU del bloque de video de la CU. Los datos residuales de una CU pueden incluir bloques de video residuales 2D que corresponden a diferentes componentes de muestra de las muestras en el bloque de video de la CU. Por ejemplo, los datos residuales pueden incluir un bloque de video residual que corresponde a las diferencias entre los componentes de luminancia de las muestras en los bloques de video predichos de las PU de la CU y los componentes de luminancia de las muestras en el bloque de video original de la CU. Además, los datos residuales de la CU pueden incluir bloques de video residuales que corresponden a las diferencias entre los componentes de crominancia de las muestras en los bloques de video predichos de las PU de la CU y los componentes de crominancia de las muestras en el bloque de video original de la CU.

El módulo 100 de predicción puede realizar la partición de árbol cuádruple para la partición de los bloques de video residuales de una CU en subbloques. Cada bloque de video residual no dividido puede estar asociado con una TU diferente de la CU. Los tamaños y posiciones de los bloques de video residuales asociados con las TU de una CU pueden basarse o no en los tamaños y posiciones de los bloques de video asociados con las PU de la CU. Una estructura de árbol cuádruple conocida como "árbol cuádruple residual" (RQT) puede incluir nodos asociados con cada uno de los bloques de video residuales. Las TU de una CU pueden corresponder a nodos de hoja de la RQT.

El módulo 104 de transformación puede generar uno o más bloques de coeficientes de transformación para cada TU de una CU aplicando una o más transformaciones a un bloque de video residual asociado con la TU. Cada uno de los bloques de coeficientes de transformación puede ser una matriz 2D de coeficientes de transformación. El módulo 104 de transformación puede aplicar varias transformaciones al bloque de video residual asociado con una TU. Por ejemplo, el módulo 104 de transformación puede aplicar una transformación discreta de coseno (DCT), una transformación direccional o una transformación conceptualmente similar al bloque de video residual asociado con una TU.

Después de que el módulo 104 de transformación genera un bloque del coeficiente de transformación asociado con una TU, el módulo 106 de cuantificación puede cuantificar los coeficientes de transformación en el bloque del coeficiente de transformación. El módulo 106 de cuantificación puede cuantificar un bloque del coeficiente de transformación asociado con una TU de una CU basándose en un valor de QP asociado con la CU.

El codificador 20 de video puede asociar un valor de QP con una CU de varias maneras. Por ejemplo, el codificador 20 de video puede realizar un análisis de distorsión de tasa en un bloque de árbol asociado con la CU. En el análisis de distorsión de tasa, el codificador 20 de video puede generar múltiples representaciones codificadas del bloque de árbol realizando una operación de codificación varias veces en el bloque de árbol. El codificador 20 de video puede

asociar diferentes valores de QP con la CU cuando el codificador 20 de video genera diferentes representaciones codificadas del bloque de árbol. El codificador 20 de video puede señalar que un valor de QP dado está asociado con la CU cuando el valor de QP dado está asociado con la CU en una representación codificada del bloque de árbol que tiene la tasa de bits y la métrica de distorsión más bajas.

El módulo 108 de cuantificación inversa y el módulo 110 de transformación inversa pueden aplicar cuantificación inversa y transformaciones inversas al bloque del coeficiente de transformación, respectivamente, para reconstruir un bloque de video residual a partir del bloque del coeficiente de transformación. El módulo 112 de reconstrucción puede agregar el bloque de video residual reconstruido a las muestras correspondientes de uno o más bloques de video predichos generados por el módulo 100 de predicción para producir un bloque de video reconstruido asociado con una TU. Al reconstruir bloques de video para cada TU de una CU de esta manera, el codificador 20 de video puede reconstruir el bloque de video de la CU.

Después de que el módulo 112 de reconstrucción reconstruya el bloque de video de una CU, el módulo 113 de filtro puede realizar una operación de desbloqueo para reducir los artefactos de bloqueo en el bloque de video asociado con la CU. Después de realizar una o más operaciones de desbloqueo, el módulo 113 de filtro puede almacenar el bloque de video reconstruido de la CU en el búfer 114 de imagen decodificada. El módulo 122 de estimación de movimiento y el módulo 124 de compensación de movimiento pueden usar una imagen de referencia que contiene el bloque de video reconstruido para realizar interpredicción en PU de imágenes posteriores. Además, el módulo 126 de intrapredicción puede usar bloques de video reconstruidos en el búfer 114 de imagen decodificada para realizar la intrapredicción en otras PU en la misma imagen que la CU.

El módulo 116 de codificación de entropía puede recibir datos de otros componentes funcionales del codificador 20 de video. Por ejemplo, el módulo 116 de codificación de entropía puede recibir bloques de coeficientes de transformación del módulo 106 de cuantificación y puede recibir elementos de sintaxis del módulo 100 de predicción. Cuando el módulo 116 de codificación de entropía recibe los datos, el módulo 116 de codificación de entropía puede realizar una o más operaciones de codificación de entropía para generar datos codificados por entropía. Por ejemplo, el codificador 20 de video puede realizar una operación de codificación de longitud variable adaptable al contexto (CAVLC), una operación CABAC, una operación de codificación de longitud variable a variable (V2V), una operación de codificación aritmética binaria adaptable al contexto basada en sintaxis (SBAC), una operación de codificación de entropía de partición de intervalo de probabilidad (PIPE) u otro tipo de operación de codificación de entropía en los datos. El módulo 116 de codificación de entropía puede emitir un flujo de bits que incluye los datos codificados por entropía.

Como parte de la realización de una operación de codificación de entropía en los datos, el módulo 116 de codificación de entropía puede seleccionar un modelo de contexto. Si el módulo 116 de codificación de entropía está realizando una operación CABAC, el modelo de contexto puede indicar estimaciones de probabilidades de bins particulares que tienen valores particulares. En el contexto de CABAC, el término "bin" se usa para referirse a un bit de una versión binarizada de un elemento de sintaxis.

El codificador 20 de video genera una unidad NAL de la porción codificada para cada porción de la imagen actual. La unidad NAL de la porción codificada para una porción puede incluir un encabezado de la porción y datos de la porción. Los datos de la porción incluyen una pluralidad de segmentos. Cada uno de los segmentos incluye bloques de árbol codificados asociados con una partición de imagen diferente. En las reivindicaciones independientes, la imagen se particiona en una pluralidad de ondas WPP. El codificador 20 de video rellena los segmentos de tal manera que cada uno de los segmentos comience en un límite de byte dentro de los datos de la porción. Por ejemplo, los segmentos en una unidad NAL de la porción codificada incluyen un segmento dado. En este ejemplo, el codificador 20 de video genera la unidad NAL de la porción codificada, al menos en parte, realizando una operación de relleno que agrega bits al segmento dado si un siguiente bloque de árbol está dentro de la porción actual y está asociado con una partición de imagen diferente a la del segmento dado.

De acuerdo con la invención, el codificador 20 de video genera el encabezado de la porción de una unidad NAL de la porción codificada de manera que el encabezado de la porción indica puntos de entrada para los segmentos en los datos de la porción de la unidad NAL de la porción codificada. Los puntos de entrada indican los desplazamientos de los bytes dentro de los datos de la porción de los segmentos. Los puntos de entrada indican desplazamientos de bytes de los segmentos. Los desplazamientos de bytes pueden ser relativos al primer bit de la unidad NAL de la porción codificada, el primer bit de los datos de la porción u otro bit en la unidad NAL de la porción codificada.

En algunos ejemplos, el codificador 20 de video puede determinar si una bandera tiene un primer valor (por ejemplo, 1). Si la bandera tiene el primer valor, el codificador 20 de video rellena uno o más de los segmentos de manera que cada segmento comience en un límite de byte. Cuando la bandera tiene un segundo valor (por ejemplo, 0), el codificador 20 de video no rellena los segmentos. Como resultado, los segmentos pueden comenzar o no en posiciones alineadas con bytes. En tales ejemplos, un conjunto de parámetros de secuencia, un conjunto de parámetros de imagen, un conjunto de parámetros de adaptación o un encabezado de la porción pueden incluir la bandera. Así, en algunos ejemplos, el codificador 20 de video puede generar un conjunto de parámetros asociados con la imagen actual, incluyendo el conjunto de parámetros una bandera. Cuando la bandera tiene un primer valor,

uno o más de los segmentos se rellenan de modo que los segmentos comiencen en los límites de byte. Cuando la bandera tiene un segundo valor, los segmentos pueden o no comenzar en los límites de bytes.

Además, en algunos ejemplos, el codificador 20 de video puede particionar la imagen actual en una pluralidad de mosaicos. Si el codificador 20 de video permite la predicción en la imagen a través de los límites de los mosaicos (es decir, cuando dos o más mosaicos dependen entre sí), el codificador 20 de video no rellena los segmentos. Como resultado, los segmentos pueden comenzar o no en posiciones alineadas con bytes. Sin embargo, si el codificador 20 de video no permite la predicción en la imagen a través de los límites de mosaico, el codificador 20 de video puede rellena uno o más de los segmentos de modo que cada uno de los segmentos comience en un límite de byte. Por lo tanto, el codificador 20 de video puede generar una unidad NAL de la porción codificada al menos en parte realizando una operación de relleno que asegura que los segmentos comiencen en los límites de byte solo después de determinar que los mosaicos son independientes entre sí. Las reivindicaciones independientes mencionan que la imagen se particiona en una pluralidad de ondas WPP. Las reivindicaciones independientes no mencionan la partición en una pluralidad de mosaicos. La Figura 4 es un diagrama de flujo que ilustra un ejemplo de operación 200 para generar datos de la porción para una porción. Como parte de la operación, si los mosaicos de la imagen actual no son independientes o el siguiente bloque de árbol no está en un mosaico diferente al bloque de árbol actual ("NO" de 208), el codificador de video puede determinar si la imagen actual se está codificando usando WPP y el siguiente bloque de árbol de la porción actual está en una onda WPP diferente que el bloque de árbol actual de la porción (210) actual. El ejemplo de operación 200 procesa una imagen que puede particionarse en mosaicos y en ondas WPP.

La Fig. 3 es un diagrama de bloques que ilustra un ejemplo de un decodificador 30 de video que está configurado para implementar las técnicas de esta divulgación. La Fig. 3 se proporciona con fines de explicación y no limita las técnicas ejemplificadas y descritas ampliamente en esta divulgación. A efectos de explicación, esta divulgación describe el decodificador 30 de video en el contexto de la codificación HEVC. Sin embargo, las técnicas de esta divulgación pueden ser aplicables a otros estándares o métodos de codificación.

En el ejemplo de la Fig. 3, el decodificador 30 de video incluye una pluralidad de componentes funcionales. Los componentes funcionales del decodificador 30 de video incluyen un módulo 150 de decodificación de entropía, un módulo 152 de predicción, un módulo 154 de cuantificación inversa, un módulo 156 de transformación inversa, un módulo 158 de reconstrucción, un módulo 159 de filtro y un búfer 160 de imagen decodificada. El módulo 152 de predicción incluye un módulo 162 de compensación de movimiento y un módulo 164 de interpredicción. En algunos ejemplos, el decodificador 30 de video puede realizar un pase de decodificación generalmente recíproco al pase de codificación descrito con respecto al codificador 20 de video de la Fig. 2. En otros ejemplos, el decodificador 30 de video puede incluir más, menos o diferentes componentes funcionales.

El decodificador 30 de video puede recibir un flujo de bits que comprende datos de video codificados. El flujo de bits puede incluir una pluralidad de elementos de sintaxis. Cuando el decodificador 30 de video recibe el flujo de bits, el módulo 150 de decodificación de entropía puede realizar una operación de análisis en el flujo de bits. Como resultado de realizar la operación de análisis en el flujo de bits, el módulo 150 de decodificación de entropía puede extraer elementos de sintaxis del flujo de bits. Como parte de la realización de la operación de análisis, el módulo 150 de decodificación de entropía puede decodificar por entropía elementos de sintaxis codificados por entropía en el flujo de bits. El módulo 152 de predicción, el módulo 154 de cuantificación inversa, el módulo 156 de transformación inversa, el módulo 158 de reconstrucción y el módulo 159 de filtro pueden realizar una operación de reconstrucción que genera datos de video decodificados basados en los elementos de sintaxis extraídos del flujo de bits.

Como se discutió anteriormente, el flujo de bits puede comprender una serie de unidades NAL. Las unidades NAL del flujo de bits pueden incluir unidades NAL del conjunto de parámetros de secuencia, unidades NAL del conjunto de parámetros de imagen, unidades NAL SEI, etc. Como parte de la realización de la operación de análisis en el flujo de bits, el módulo 150 de decodificación de entropía puede realizar operaciones de análisis que extraen y decodifican por entropía conjuntos de parámetros de secuencia de unidades NAL de conjunto de parámetros de secuencia, conjuntos de parámetros de imagen de unidades NAL del conjunto de parámetros de imagen, datos SEI de unidades NAL SEI, etc.

Además, las unidades NAL del flujo de bits pueden incluir unidades NAL de la porción codificada. Como parte de la realización de la operación de análisis en el flujo de bits, el decodificador 30 de video puede realizar operaciones de análisis que extraen y decodifican por entropía segmentos codificados de las unidades NAL de las porciones codificadas. Cada una de las porciones codificadas puede incluir un encabezado de la porción y datos de la porción. El encabezado de la porción puede contener elementos de sintaxis pertenecientes a una porción. Los elementos de sintaxis en el encabezado de la porción pueden incluir un elemento de sintaxis que identifica un conjunto de parámetros de imagen asociado con una imagen que contiene la porción.

Los datos de la porción de una unidad NAL de la porción codificada pueden incluir múltiples segmentos. Cada uno de los segmentos incluye bloques de árbol codificados asociados con una partición de imagen diferente (por ejemplo, un mosaico o una onda WPP). Las reivindicaciones independientes mencionan que la imagen se particiona en una pluralidad de ondas WPP. Uno o más de los segmentos en los datos de la porción se rellenan de modo que cada uno de los segmentos comience en un límite de byte. El encabezado de la porción de la unidad NAL de la porción codificada

indica puntos de entrada para los segmentos. En este caso, debido a que los segmentos siempre comienzan en los límites de los bytes, el decodificador 30 de video puede asignar diferentes segmentos a diferentes subprocesos de decodificación de una manera simple usando direccionamiento de memoria por bytes. Los diferentes subprocesos de decodificación pueden analizar los bloques de árbol codificados de los segmentos y reconstruir los datos de video asociados con los bloques de árbol correspondientes en paralelo.

Como parte de la extracción de datos de la porción de unidades NAL de la porción codificada, el módulo 150 de decodificación de entropía puede realizar operaciones de análisis que extraen elementos de sintaxis de las CU codificadas. Los elementos de sintaxis extraídos pueden incluir elementos de sintaxis asociados con bloques de coeficientes de transformación. El módulo 150 de decodificación de entropía puede entonces realizar operaciones de decodificación de CABAC en algunos de los elementos de sintaxis.

Después de que el módulo 150 de decodificación de entropía realiza una operación de análisis en una CU no particionada, el decodificador 30 de video puede realizar una operación de reconstrucción en la CU no particionada. Para realizar la operación de reconstrucción en una CU no particionada, el decodificador 30 de video puede realizar una operación de reconstrucción en cada TU de la CU. Al realizar la operación de reconstrucción para cada TU de la CU, el decodificador 30 de video puede reconstruir un bloque de video residual asociado con la CU.

Como parte de la realización de una operación de reconstrucción en una TU, el módulo 154 de cuantificación inversa puede cuantificar inversamente, es decir, descuantificar, un bloque del coeficiente de transformación asociado con la TU. El módulo 154 de cuantificación inversa puede cuantificar inversamente el bloque del coeficiente de transformación de una manera similar a los procesos de cuantificación inversa propuestos para HEVC o definidos por el estándar de decodificación H.264. El módulo 154 de cuantificación inversa puede usar un parámetro de cuantificación QP calculado por el codificador 20 de video para una CU del bloque del coeficiente de transformación para determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa para aplicar el módulo 154 de cuantificación inversa.

Después de que el módulo 154 de cuantificación inversa cuantifica inversamente un bloque del coeficiente de transformación, el módulo 156 de transformación inversa puede generar un bloque de vídeo residual para la TU asociada con el bloque del coeficiente de transformación. El módulo 156 de transformación inversa puede aplicar una transformación inversa al bloque del coeficiente de transformación para generar el bloque de vídeo residual para la TU. Por ejemplo, el módulo 156 de transformación inversa puede aplicar una DCT inversa, una transformación entera inversa, una transformación de Karhunen-Loeve inversa (KLT), una transformación rotacional inversa, una transformación direccional inversa u otra transformación inversa al bloque del coeficiente de transformación.

En algunos ejemplos, el módulo 156 de transformación inversa puede determinar una transformación inversa para aplicar al bloque del coeficiente de transformación en función de la señalización del codificador 20 de video. En tales ejemplos, el módulo 156 de transformación inversa puede determinar la transformación inversa en función de una transformación señalada en el nodo raíz de un árbol cuádruple para un bloque de árbol asociado con el bloque del coeficiente de transformación. En otros ejemplos, el módulo 156 de transformación inversa puede inferir la transformación inversa a partir de una o más características de codificación, tales como el tamaño del bloque, el modo de codificación o similares. En algunos ejemplos, el módulo 156 de transformación inversa puede aplicar una transformación inversa en cascada.

En algunos ejemplos, el módulo 162 de compensación de movimiento puede refinar el bloque de video predicho de una PU realizando una interpolación basada en filtros de interpolación. En los elementos sintácticos pueden incluirse identificadores de filtros de interpolación que se utilizarán para la compensación de movimiento con precisión de submuestra. El módulo 162 de compensación de movimiento puede usar los mismos filtros de interpolación usados por el codificador 20 de video durante la generación del bloque de video predicho de la PU para calcular valores interpolados para muestras de subenteros de un bloque de referencia. El módulo 162 de compensación de movimiento puede determinar los filtros de interpolación usados por el codificador 20 de video de acuerdo con la información de sintaxis recibida y usar los filtros de interpolación para producir el bloque de video predicho.

Si una PU está codificada usando intrapredicción, el módulo 164 de intrapredicción puede realizar intrapredicción para generar un bloque de video predicho para la PU. Por ejemplo, el módulo 164 de intrapredicción puede determinar un modo de intrapredicción para la PU basándose en elementos de sintaxis en el flujo de bits. El flujo de bits puede incluir elementos de sintaxis que el módulo 164 de intrapredicción puede usar para determinar el modo de intrapredicción de la PU.

En algunos casos, los elementos de sintaxis pueden indicar que el módulo 164 de intrapredicción va a utilizar el modo de intrapredicción de otra PU para determinar el modo de intrapredicción de la PU actual. Por ejemplo, puede ser probable que el modo de intrapredicción de la PU actual sea el mismo que el modo de intrapredicción de una PU vecina. En otras palabras, el modo de intrapredicción de la PU vecina puede ser el modo más probable para la PU actual. Por lo tanto, en este ejemplo, el flujo de bits puede incluir un pequeño elemento de sintaxis que indica que el modo de intrapredicción de la PU es el mismo que el modo de intrapredicción de la PU vecina. El módulo 164 de intrapredicción puede entonces usar el modo de intrapredicción para generar datos de predicción (por ejemplo, muestras predichas) para la PU basándose en los bloques de vídeo de las PU espacialmente vecinas.



El módulo 158 de reconstrucción puede usar los bloques de video residuales asociados con las TU de una CU y los bloques de video predichos de las PU de la CU, es decir, datos de intrapredicción o datos de interpredicción, según corresponda, para reconstruir el bloque de video de la CU. Por lo tanto, el decodificador 30 de video puede generar un bloque de video predicho y un bloque de video residual basado en elementos de sintaxis en el flujo de bits y puede generar un bloque de video basado en el bloque de video predicho y el bloque de video residual.

Después de que el módulo 158 de reconstrucción reconstruya el bloque de video de la CU, el módulo 159 de filtro puede realizar una operación de desbloqueo para reducir los artefactos de bloqueo asociados con la CU. Después de que el módulo 159 de filtro realiza una operación de desbloqueo para reducir los artefactos de bloqueo asociados con la CU, el decodificador 30 de video puede almacenar el bloque de video de la CU en el búfer 160 de la imagen decodificada. El búfer 160 de la imagen decodificada puede proporcionar imágenes de referencia para la compensación de movimiento subsiguiente, la interpredicción, y la presentación en un dispositivo de visualización, tal como el dispositivo 32 de visualización de la Fig. 1. Por ejemplo, el decodificador 30 de video puede realizar, con base en los bloques de video en el búfer 160 de la imagen decodificada, operaciones de intrapredicción o interpredicción en las PU de otras CU.

La Fig. 4 es un diagrama de flujo que ilustra un ejemplo de la operación 200 para generar datos de una porción para una porción. Un codificador de video, tal como el codificador 20 de video (Figs. 1 y 2), puede realizar la operación 200. El ejemplo de la Fig. 4 es simplemente un ejemplo. Otras operaciones de ejemplo pueden generar datos de la porción de otras maneras.

Después de que el codificador de video comience la operación 200, el codificador de video puede inicializar una dirección de bloque de árbol de modo que la dirección de bloque de árbol identifique un bloque de árbol inicial de una porción (202) actual. La porción actual puede ser una porción que el codificador de video está codificando actualmente. El bloque de árbol inicial de la porción actual puede ser el primer bloque de árbol asociado con la porción actual de acuerdo con un orden de codificación de bloques de árbol para la imagen actual. Para facilitar la explicación, esta divulgación puede hacer referencia al bloque de árbol identificado por la dirección del bloque de árbol como el bloque de árbol actual.

El codificador de video puede agregar elementos de sintaxis para el bloque de árbol actual a los datos de la porción de una unidad NAL de la porción codificada para la porción (204) actual. Los elementos de sintaxis para el bloque de árbol actual pueden incluir elementos de sintaxis en el árbol cuádruple del bloque de árbol actual. Los elementos de sintaxis en el árbol cuádruple del bloque de árbol actual pueden incluir elementos de sintaxis que indican modos de intrapredicción, información de movimiento, elementos de sintaxis que indican niveles del coeficiente de transformación, etc.

Además, el codificador de video puede determinar si hay más datos en la porción (206) actual. Puede haber más datos en la porción actual si el bloque de árbol indicado por la dirección del bloque de árbol está dentro de la porción actual. En respuesta a determinar que no hay más datos en la porción actual ("NO" de 206), el codificador de video puede finalizar la operación 200 porque el codificador de video ha agregado todos los elementos de sintaxis necesarios a los datos de la porción.

El codificador de video puede determinar si hay más datos en la porción actual de varias maneras. Por ejemplo, el codificador de video puede invocar una función "arbol\_de codificación ()" para generar los elementos de sintaxis para un bloque de árbol. En este ejemplo, la función "arbol\_de codificación ()" puede devolver una "Bandera de más Datos" que indica si hay más datos en la porción actual.

En respuesta a la determinación de que hay más datos asociados con la porción actual ("Sí" de 206), el codificador de video puede determinar si los mosaicos de la imagen actual son independientes y si el siguiente bloque de árbol de la porción actual está en un mosaico diferente al bloque de árbol actual de la porción (208) actual. Como se describió anteriormente, los mosaicos de una imagen pueden ser independientes si se prohíbe la predicción dentro de la imagen (p. ej., intrapredicción, interpredicción utilizando datos de la imagen actual y selección de contexto de CABAC con base en los datos de otros mosaicos de la imagen actual). El codificador de video puede determinar si los mosaicos de la imagen actual son independientes de varias maneras. Por ejemplo, un conjunto de parámetros de secuencia asociado con la imagen actual puede incluir un elemento de sintaxis "idc\_de independencia\_del límite\_del mosaico". En este ejemplo, si "idc\_de independencia\_del límite\_del mosaico" es igual a 0, los mosaicos de la imagen actual no son independientes y se permite la predicción en la imagen a través de los límites de los mosaicos. Si "idc\_de independencia\_del límite\_del mosaico" es igual a 0, la predicción en la imagen a través de los límites de las porciones aún pueden estar prohibidos. Si "idc\_de independencia\_del límite\_del mosaico" es igual a 1, los mosaicos de la imagen actual son independientes y no se permite la predicción en la imagen a través de los límites de los mosaicos.

El codificador de video puede determinar de varias maneras si el siguiente bloque de árbol de la porción actual está en un mosaico diferente que el bloque de árbol actual de la porción actual. Por ejemplo, el codificador de video puede determinar la dirección del bloque de árbol del siguiente bloque de árbol de la porción actual. En este ejemplo, el codificador de video puede invocar una función "Nuevo Mosaico(...)" que toma la dirección del bloque de árbol del

siguiente bloque de árbol como un parámetro y devuelve un valor "nueva Bandera de Mosaico" que indica si el siguiente bloque de árbol está en un mosaico diferente que el bloque de árbol actual.

Si los mosaicos de la imagen actual no son independientes o el siguiente bloque de árbol no está en un mosaico diferente al bloque de árbol actual ("NO" de 208), el codificador de video puede determinar si la imagen actual se está codificando usando WPP y el siguiente bloque de árbol de la porción actual está en una onda WPP diferente que el bloque de árbol actual de la porción (210) actual. El codificador de video puede determinar de varias maneras si el siguiente bloque de árbol de la porción actual está en una onda WPP diferente que el bloque de árbol actual de la porción actual. Por ejemplo, el codificador de video puede determinar la dirección del bloque de árbol del siguiente bloque de árbol de la porción actual. En este ejemplo, el codificador de video puede invocar una función "Nueva Onda(...)" que toma la dirección del bloque de árbol del siguiente bloque de árbol como parámetro y devuelve un valor "nueva Bandera de Onda" que indica si el siguiente bloque de árbol está en una onda WPP diferente que el bloque de árbol actual.

En respuesta a la determinación de que la imagen actual se está codificando mediante WPP y el siguiente bloque de árbol está en una onda WPP diferente a la del bloque de árbol actual ("Sí" de 210) o en respuesta a la determinación de que los mosaicos de la imagen actual son independiente y el siguiente bloque de árbol está en un mosaico diferente al bloque de árbol actual ("Sí" de 208), el codificador de video puede determinar si el segmento actual tenga los bytes alineados (212). En otras palabras, el codificador de video puede determinar si el segmento actual termina en un límite de byte. El segmento actual es el segmento asociado con la partición de imagen (por ejemplo, mosaico u onda WPP) con el que está asociado el bloque de árbol actual. En respuesta a la determinación de que el segmento actual no tenga los bytes alineados ("NO" de 212), el codificador de video puede agregar un bit de relleno al final del segmento actual (214). El bit de relleno puede tener varios valores. Por ejemplo, el bit de relleno siempre puede tener un valor igual a 1. En otros ejemplos, el bit de relleno siempre puede tener un valor igual a 0.

Después de añadir el bit de relleno al final del segmento actual, el codificador de video puede determinar de nuevo si el segmento actual tenga los bytes alineados (212). De esta forma, el codificador de video puede continuar agregando bits de relleno al final de los datos de la porción hasta que el segmento actual tenga los bytes alineados.

En respuesta a la determinación de que los datos de la porción tengan los bytes alineados ("Sí" de 212), el codificador de video puede actualizar la dirección del bloque de árbol (216). El codificador de video puede actualizar la dirección del bloque de árbol de manera que la dirección del bloque de árbol indique el siguiente bloque de árbol de acuerdo con un orden de codificación de bloque de árbol de la imagen actual. Por ejemplo, cuando el codificador de video actualiza la dirección del bloque de árbol, la dirección del bloque de árbol puede identificar un bloque de árbol a la derecha del bloque de árbol indicado previamente por la dirección del bloque de árbol. La Fig. 7, que se describe en detalle a continuación, es un diagrama conceptual que ilustra un ejemplo de orden de codificación de bloques de árbol para una imagen que se particiona en múltiples mosaicos.

Después de actualizar la dirección del bloque de árbol, el codificador de video puede determinar si hay más datos en la porción (218) actual. En respuesta a la determinación de que hay más datos en la porción actual ("Sí" de 218) o en respuesta a la determinación de que la imagen actual no se está codificando mediante WPP y el siguiente bloque de árbol no está en un mosaico diferente al del bloque de árbol actual ("NO" de 210), el codificador de video puede agregar los elementos de sintaxis para el bloque de árbol actual a los datos de la porción (204). De esta forma, el codificador de video puede agregar los elementos de sintaxis para cada bloque de árbol de la porción actual a los datos de la porción y puede garantizar que los segmentos asociados con diferentes particiones de imagen se rellenen de manera que los segmentos comiencen en los límites de byte.

En respuesta a la determinación de que no hay más datos en la porción actual ("NO" de 218), el codificador de video puede finalizar la operación 200 porque el codificador de video puede haber agregado todos los elementos de sintaxis de la porción actual a los datos de la porción.

La Fig. 5 es un diagrama de flujo que ilustra un ejemplo de operación 250 para decodificar una unidad NAL de la porción codificada. Estas operaciones se pueden realizar como parte de la decodificación de una imagen en las reivindicaciones 2 o 5. Un decodificador de video, tal como el decodificador 30 de video (Figs. 1 y 3), puede realizar la operación 250. El ejemplo de la Fig. 5 es simplemente un ejemplo. Otras operaciones de ejemplo pueden realizar otras operaciones para decodificar unidades NAL de la porción codificada.

En el ejemplo de la Fig. 5, el decodificador de video puede almacenar una unidad NAL de la porción codificada en la memoria (252) direccionada por bytes. La unidad NAL de la porción codificada puede incluir un encabezado de la porción y datos de la porción. Los datos de la porción pueden incluir una pluralidad de segmentos. Uno o más de los segmentos se pueden rellenar de manera que cada segmento comience en un límite de byte.

Después de almacenar la unidad NAL de la porción codificada en la memoria, el decodificador de video puede identificar las posiciones de los segmentos dentro de los datos de la porción de la unidad (254) NAL de la porción codificada. El decodificador de video puede identificar las posiciones de los segmentos de varias maneras. Por ejemplo, el decodificador de video puede identificar las posiciones de los segmentos basándose en elementos de

sintaxis en el encabezado de la porción de la unidad NAL de la porción codificada que indica desplazamientos de bytes de los segmentos. En este ejemplo, es posible que el encabezado de la porción no incluya un desplazamiento de bytes para el primer segmento de los datos de la porción porque la posición del primer segmento puede seguir inmediatamente al final del encabezado de la porción. En otro ejemplo, el decodificador de video puede identificar las posiciones de los segmentos basándose en marcadores de puntos de entrada en los datos de la porción. Los marcadores de puntos de entrada pueden ser valores dispuestos entre los segmentos.

Después de identificar las posiciones de los segmentos dentro de los datos de la porción, el decodificador de video puede asignar dos o más de los segmentos a dos o más subprocesos (256) de decodificación diferentes. Cada uno de los subprocesos de decodificación puede analizar los elementos de sintaxis de los bloques de árbol codificados en el segmento asignado al subproceso de decodificación y reconstruir los bloques de video para los bloques de árbol correspondientes como se ha descrito anteriormente.

La Fig. 6 es un diagrama conceptual que ilustra el procesamiento paralelo de frente de onda. Las reivindicaciones independientes mencionan que la imagen se particiona en una pluralidad de ondas WPP. Como se ha descrito anteriormente, una imagen puede particionarse en bloques de vídeo, cada uno de los cuales está asociado a un bloque de árbol. La Fig. 6 ilustra los bloques de video asociados con los bloques de árbol como una cuadrícula de cuadrados blancos. La imagen incluye filas de bloques de árbol 300A-300E (colectivamente, "filas 300 de bloques de árbol").

Un primer subproceso puede codificar bloques de árbol en la fila 300A de bloques de árbol. Al mismo tiempo, otros subprocesos pueden estar codificando bloques de árbol en las filas 300B, 300C y 300D de bloques de árbol. En el ejemplo de la Fig. 6, el primer subproceso codifica actualmente un bloque de árbol 302A, un segundo subproceso codifica actualmente un bloque de árbol 302B, un tercer subproceso codifica actualmente un bloque de árbol 302C y un cuarto subproceso codifica actualmente un bloque de árbol 302D. Esta divulgación puede hacer referencia a los bloques de árbol 302A, 302B, 302C y 302D colectivamente como "bloques de árbol 302 actuales". Debido a que el codificador de video puede comenzar a codificar una fila de bloques de árbol después de que se hayan codificado más de dos bloques de árbol de una fila inmediatamente superior, los bloques de árbol 302 actuales se desplazan horizontalmente entre sí por los anchos de dos bloques de árbol.

En el ejemplo de la Fig. 6, los subprocesos pueden usar datos de los bloques de árbol indicados por las flechas grises gruesas al realizar la intrapredicción o la interpredicción para las CU en los bloques de árbol 302 actuales. (Cuando los subprocesos realizan la interpredicción para las CU, los subprocesos también pueden usar datos de una o más marcos de referencia). Cuando un subproceso codifica un bloque de árbol dado, el subproceso puede seleccionar uno o más contextos de CABAC en función de la información asociada con bloques de árbol codificados previamente. El subproceso puede usar uno o más contextos de CABAC para realizar la codificación de CABAC en elementos de sintaxis asociados con la primera CU del bloque de árbol dado. Si el bloque de árbol dado no es el bloque de árbol más a la izquierda de una fila, el subproceso puede seleccionar uno o más contextos de CABAC en función de la información asociada con una última CU del bloque de árbol a la izquierda del bloque de árbol dado. Si el bloque de árbol dado es el bloque de árbol más a la izquierda de una fila, el subproceso puede seleccionar uno o más contextos de CABAC en función de la información asociada con una última CU de un bloque de árbol que está arriba y dos bloques de árbol a la derecha del bloque de árbol dado. Los subprocesos pueden usar datos de las últimas CU de los bloques de árbol indicados por las flechas negras delgadas para seleccionar contextos de CABAC para las primeras CU de los bloques de árbol 302 actuales.

La Fig. 7 es un diagrama conceptual que ilustra un ejemplo de orden de codificación de bloques de árbol para una imagen 350 que se particiona en múltiples mosaicos 352A, 352B y 352C. Cada bloque blanco cuadrado en la imagen 350 representa un bloque de video asociado con un bloque de árbol. Las líneas punteadas verticales gruesas indican límites de mosaico verticales de ejemplo. La línea gris gruesa indica un ejemplo de un límite de la porción. Las reivindicaciones independientes mencionan que la imagen se particiona en una pluralidad de ondas WPP. Las reivindicaciones independientes no mencionan la partición en una pluralidad de mosaicos. La Fig. 4 indica que una imagen puede codificarse usando WPP y mosaicos.

Los números en los bloques de video indican posiciones de los bloques de árbol correspondientes (LCU) en un orden de codificación de bloques de árbol para la imagen 350. Como se ilustra en el ejemplo de la Fig. 7, cada uno de los bloques de árbol en el mosaico 352A más a la izquierda aparece en el orden de codificación de bloques de árbol antes que cualquier bloque de árbol en el mosaico 352B central. Cada uno de los bloques de árbol en el mosaico 352B central aparece en el orden de codificación de bloques de árbol antes que cualquier bloque de árbol en el mosaico 352C más a la derecha. Dentro de cada uno de los mosaicos 352A, 352B y 352C, los bloques de árbol están codificados de acuerdo con un orden de escaneo de trama.

Un codificador de video puede generar dos unidades NAL de porción codificada para la imagen 350. La primera unidad NAL de la porción codificada puede estar asociada con la porción izquierda de la imagen 350. La primera unidad NAL de la porción codificada puede incluir representaciones codificadas de bloques de árbol 1-23. Los datos de la porción de la primera unidad NAL de la porción codificada pueden incluir dos segmentos. El primer segmento puede incluir las representaciones codificadas de los bloques de árbol 1-15. El segundo segmento puede incluir las representaciones

codificadas de los bloques de árbol 16-30. De acuerdo con las técnicas de esta divulgación, el primer segmento puede rellenarse de manera que el segundo segmento comience en un límite de byte.

Una segunda unidad NAL de la porción codificada puede estar asociada con la porción derecha de la imagen 350. La segunda unidad NAL de la porción codificada puede incluir representaciones codificadas de bloques de árbol 24-45. Los datos de la porción de la segunda unidad NAL de la porción codificada pueden incluir dos segmentos. El primer segmento puede incluir las representaciones codificadas de los bloques de árbol 24-30. El segundo segmento puede incluir las representaciones codificadas de los bloques de árbol 31-45. El primer segmento se puede rellenar de manera que el segundo segmento comience en un límite de byte.

La Fig. 8 es un diagrama conceptual que ilustra un ejemplo de una unidad 400 NAL de la porción codificada. Como se ilustra en el ejemplo de la Fig. 8, la unidad 400 NAL de la porción codificada incluye un encabezado 402 de la porción y datos 400 de la porción. Los datos 404 de la porción incluyen un primer segmento 406 y un segundo segmento 408. El segmento 406 incluye bloques de árbol 410A-410N codificados y datos 412 de relleno. El segmento 408 incluye bloques de árbol 414A -414N codificados.

En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, firmware o cualquier combinación de los mismos. Si se implementan en software, las funciones pueden almacenarse o transmitirse, como una o más instrucciones o códigos, en un medio legible por ordenador y ser ejecutadas por una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible tal como medios de almacenamiento de datos o medios de comunicación que incluyen cualquier medio que facilite la transferencia de un programa de ordenador 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) un medio de almacenamiento tangible legible por ordenador que no es transitorio o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que puedan acceder uno o más ordenadores o uno o más procesadores para recuperar instrucciones, códigos y/o estructuras de datos para la implementación de las técnicas descritas en esta divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

A modo de ejemplo, y no de limitación, tales medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento en disco óptico, almacenamiento en disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que pueda usarse para almacenar el código de programa deseado en forma de instrucciones o estructuras de datos y que puedan ser accedidos por un ordenador. Además, cualquier conexión se denomina correctamente un medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, servidor u otra fuente remota mediante 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, el cable de fibra óptica, el par trenzado, el DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas se incluyen en la definición de medio. Debe entenderse, sin embargo, que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que están dirigidos a medios de almacenamiento tangibles no transitorios. Disco (disk) y disco (disc), como se usan en este documento, incluyen un disco compacto (CD), un disco láser, un disco óptico, un disco versátil digital (DVD), un disquete y un disco Blu-ray, donde los discos (disks) generalmente reproducen datos magnéticamente, mientras que los discos (discs) reproducen datos ópticamente con láser. Las combinaciones de lo anterior también deben incluirse dentro del alcance de los medios legibles por ordenador.

Las instrucciones pueden ser ejecutadas por uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de uso general, circuitos integrados específicos de la aplicación (ASIC), matrices lógicas programables de campo (FPGA) u otros circuitos lógicos integrados o discretos. En consecuencia, el término "procesador", como se usa en el presente documento, puede referirse a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en este documento puede proporcionarse dentro de módulos de software y/o hardware dedicados configurados para codificar y decodificar, o incorporarse en un códec combinado. Además, las técnicas podrían implementarse completamente en uno o más circuitos o elementos lógicos.

Las técnicas de esta divulgación pueden implementarse en una amplia variedad de dispositivos o aparatos, incluidos un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En esta divulgación se describen varios componentes, módulos o unidades para enfatizar los aspectos funcionales de los dispositivos configurados para realizar las técnicas divulgadas, pero no necesariamente requieren la realización por diferentes unidades de hardware. Más bien, como se describió anteriormente, varias unidades pueden combinarse en una unidad de hardware códec o proporcionarse mediante una colección de unidades de hardware interoperativas, que incluyen uno o más procesadores como se describió anteriormente, junto con software y/o firmware adecuados.

Se han descrito varios ejemplos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.

**REIVINDICACIONES**

1. Un método para codificar datos de video, comprendiendo el método:

5 generar una capa de abstracción de red de porción codificada, NAL, unidad (400) que comprende una pluralidad de bits que incluyen representaciones codificadas de una pluralidad de bloques de árbol (410A-N, 414A-N) que están asociados con una porción de una imagen (350), donde la imagen se divide en una pluralidad de porciones, incluyendo cada porción de la pluralidad de porciones un número entero de bloques de árbol; y  
 10 que codifica la imagen, caracterizado porque:

i) la imagen se codifica usando procesamiento paralelo de frente de onda, WPP, mediante el cual la imagen también se particiona en una pluralidad de ondas WPP (300A-E), cada una de las ondas WPP comprende una fila diferente de bloques de árbol en la imagen,

15 ii) la unidad NAL de la porción codificada comprende una pluralidad de bits que incluyen representaciones codificadas de una pluralidad de bloques de árbol consecutivos (410A-N, 414A-N), la pluralidad de bloques de árbol consecutivos de la unidad NAL de porción codificada incluye bloques de árbol de dos o más ondas WPP adyacentes de la imagen, iii) las representaciones codificadas de los bloques de árbol (410A-N, 414A-N) se agrupan dentro de la unidad NAL de la porción codificada en una pluralidad de segmentos (406, 408), siendo cada segmento una serie consecutiva de bits que comprende bits que codifican los bloques de árbol de una onda diferente de las dos o más ondas WPP adyacentes,

20 iv) el método comprende además rellenar (412) uno o más de los segmentos (406, 408), de modo que cada uno de los segmentos comience en un límite de byte, y

25 v) la unidad NAL de la porción codificada comprende un encabezado de la porción que indica puntos de entrada para uno o más de los segmentos, donde los puntos de entrada para los segmentos indican desplazamientos de bytes de los segmentos.

2. Un método (250) de decodificación de datos de video, comprendiendo el método:

30 almacenar (252) una capa de abstracción de red de porción codificada, NAL, unidad que comprende, una pluralidad de bits que incluyen representaciones codificadas de una pluralidad de bloques de árbol (410A-N, 414A-N) que están asociados con una porción de una imagen (350), donde la imagen se divide en una pluralidad de porciones, cada porción de la pluralidad de porciones incluye un número entero de bloques de árbol; y  
 35 que decodifica la imagen, caracterizado porque:

i) la imagen se decodifica usando procesamiento paralelo de frente de onda, WPP, mediante el cual la imagen también se particiona en una pluralidad de ondas WPP (300A-E), cada una de las ondas WPP comprende una fila diferente de bloques de árbol en la imagen,

40 ii) la unidad NAL de la porción codificada comprende una pluralidad de bits que incluyen representaciones codificadas de una pluralidad de bloques de árbol consecutivos (410A-N, 414A-N), la pluralidad de bloques de árbol consecutivos de la unidad NAL de la porción codificada incluye bloques de árbol de dos o más ondas WPP adyacentes de la imagen,

45 iii) las representaciones codificadas de los bloques de árbol se agrupan dentro de la unidad NAL de la porción codificada en una pluralidad de segmentos (406, 408), siendo cada segmento una serie consecutiva de bits que codifican los bloques de árbol de una onda diferente de las dos o más ondas WPP adyacentes,

iv) uno o más de los segmentos se rellenan (412) de manera que cada uno de los segmentos (406, 408) comience en un límite de byte,

50 v) la unidad NAL de la porción codificada comprende un encabezado de la porción que indica puntos de entrada para uno o más de los segmentos, donde los puntos de entrada para los segmentos indican desplazamientos de bytes de los segmentos, y

vi) decodificar la imagen usando WPP incluye decodificar las representaciones codificadas de los bloques de árbol en dos o más de los segmentos en paralelo usando los puntos de entrada indicados en el encabezado de la porción.

55 3. Un dispositivo de codificación de video para codificar datos de video, comprendiendo el dispositivo de codificación de video:

60 medios para generar una capa de abstracción de red de porción codificada, NAL, unidad (400) que comprende una pluralidad de bits que incluyen representaciones codificadas de una pluralidad de bloques de árbol (410A-N, 414A-N) que están asociados con una porción de una imagen (350), en donde la imagen se divide en una pluralidad de porción, incluyendo cada porción de la pluralidad de porciones un número entero de bloques de árbol; y  
 medios para codificar la imagen, caracterizado porque:

- i) la imagen se codifica usando procesamiento paralelo de frente de onda, WPP, mediante el cual la imagen también se particiona en una pluralidad de ondas WPP (300A-E), cada una de las ondas WPP comprende una fila diferente de bloques de árbol en la imagen,
- 5 ii) la unidad NAL de la porción codificada comprende una pluralidad de bits que incluyen representaciones codificadas de una pluralidad de bloques de árbol consecutivos (410A-N, 414A-N) que incluyen bloques de árbol de dos o más ondas WPP adyacentes de la imagen,
- 10 iii) las representaciones codificadas de los bloques de árbol (410A-N, 414A-N) se agrupan dentro de la unidad NAL de la porción codificada en una pluralidad de segmentos (406, 408), siendo cada segmento una serie consecutiva de bits que comprende bits que codifican los bloques de árbol de una onda diferente de las dos o más ondas WPP adyacentes,
- iv) el dispositivo comprende además medios para rellenar (412) uno o más de los segmentos (406, 408), de modo que cada uno de los segmentos comience en un límite de byte, y
- v) la unidad NAL de la porción codificada comprende un encabezado de la porción que indica puntos de entrada para uno o más de los segmentos, donde los puntos de entrada para los segmentos indican desplazamientos de bytes de
- 15 los segmentos.
4. Un dispositivo de decodificación de video para decodificar datos de video, comprendiendo el dispositivo de decodificación de video:
- 20 medios para almacenar (252) una unidad de capa de abstracción de red (NAL) de segmentos codificados que incluye representaciones codificadas de una pluralidad de bloques de árbol (410A-N, 414A-N) asociados con un segmento de la imagen, donde la imagen se divide en una pluralidad de rebanadas, incluyendo cada rebanada de la pluralidad de rebanadas un número entero de bloques de árbol; y
- 25 medios para decodificar la imagen, caracterizado porque:
- i) la imagen se decodifica usando procesamiento paralelo de frente de onda, WPP, mediante el cual la imagen también se divide en una pluralidad de ondas WPP (300A-E), cada una de las ondas WPP comprende una fila diferente de bloques de árbol en la imagen,
- 30 ii) la unidad NAL de la porción codificada comprende una pluralidad de bits que incluyen representaciones codificadas de una pluralidad de bloques de árbol consecutivos (410A-N, 414A-N), la pluralidad de bloques de árbol consecutivos de la unidad NAL de la porción codificada incluye bloques de árbol de dos o más bloques de árbol adyacentes WPP ondas de la imagen,
- 35 iii) las representaciones codificadas de los bloques de árbol se agrupan dentro de la unidad NAL de la porción codificada en una pluralidad de segmentos (406, 408), siendo cada segmento una serie consecutiva de bits que codifican los bloques de árbol de una diferente de las dos o más ondas WPP adyacentes,
- iv) uno o más de los segmentos se rellenan (412) de manera que cada uno de los segmentos (406, 408) comience en un límite de byte,
- v) la unidad NAL de la porción codificada comprende un encabezado de la porción que indica puntos de entrada para uno o más de los segmentos, donde los puntos de entrada para los segmentos indican desplazamientos de bytes de
- 40 los segmentos, y
- vi) decodificar la imagen usando WPP comprende decodificar las representaciones codificadas de los bloques de árbol en dos o más de los segmentos en paralelo usando los puntos de entrada indicados en el encabezado de la porción.
- 45 5. Un producto de programa informático que comprende uno o más medios de almacenamiento legibles por ordenador que almacenan instrucciones que, cuando son ejecutadas por uno o más procesadores, configuran un dispositivo para llevar a cabo un método de acuerdo con la reivindicación 1 o la reivindicación 2.

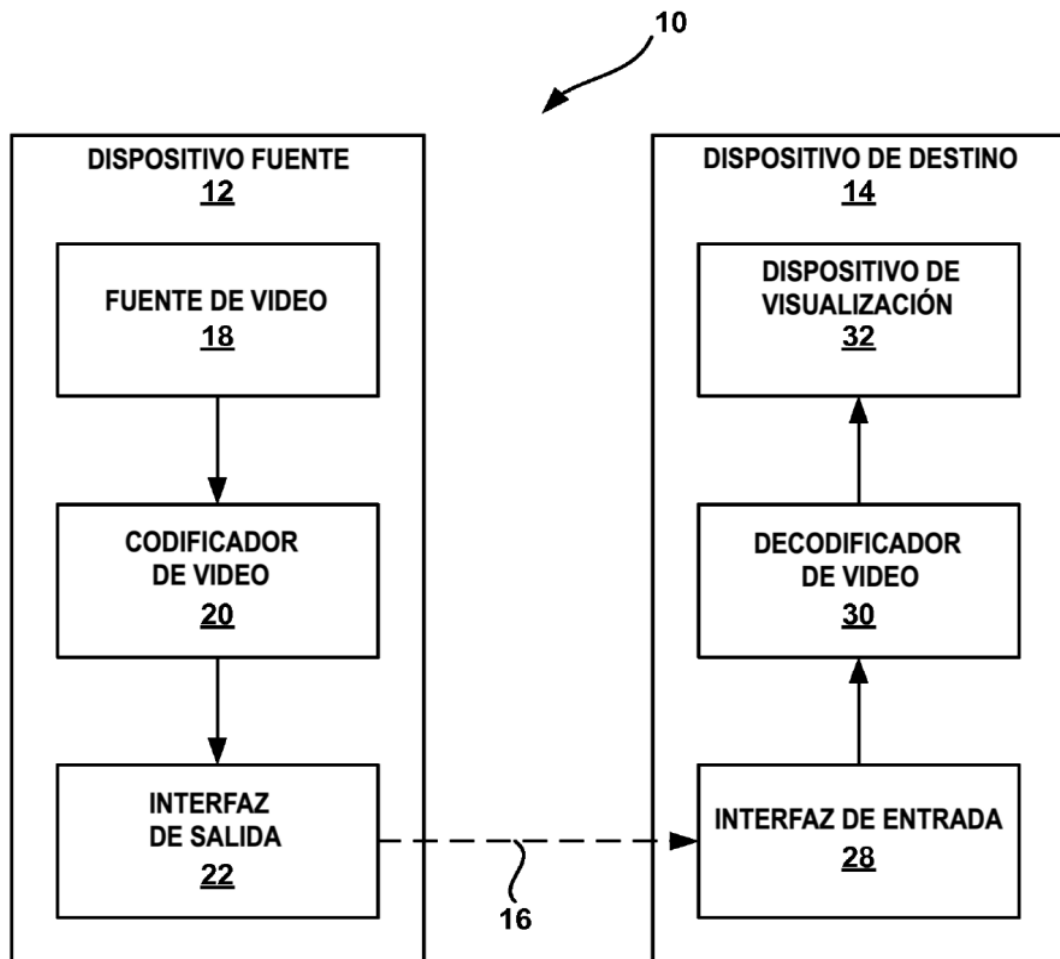
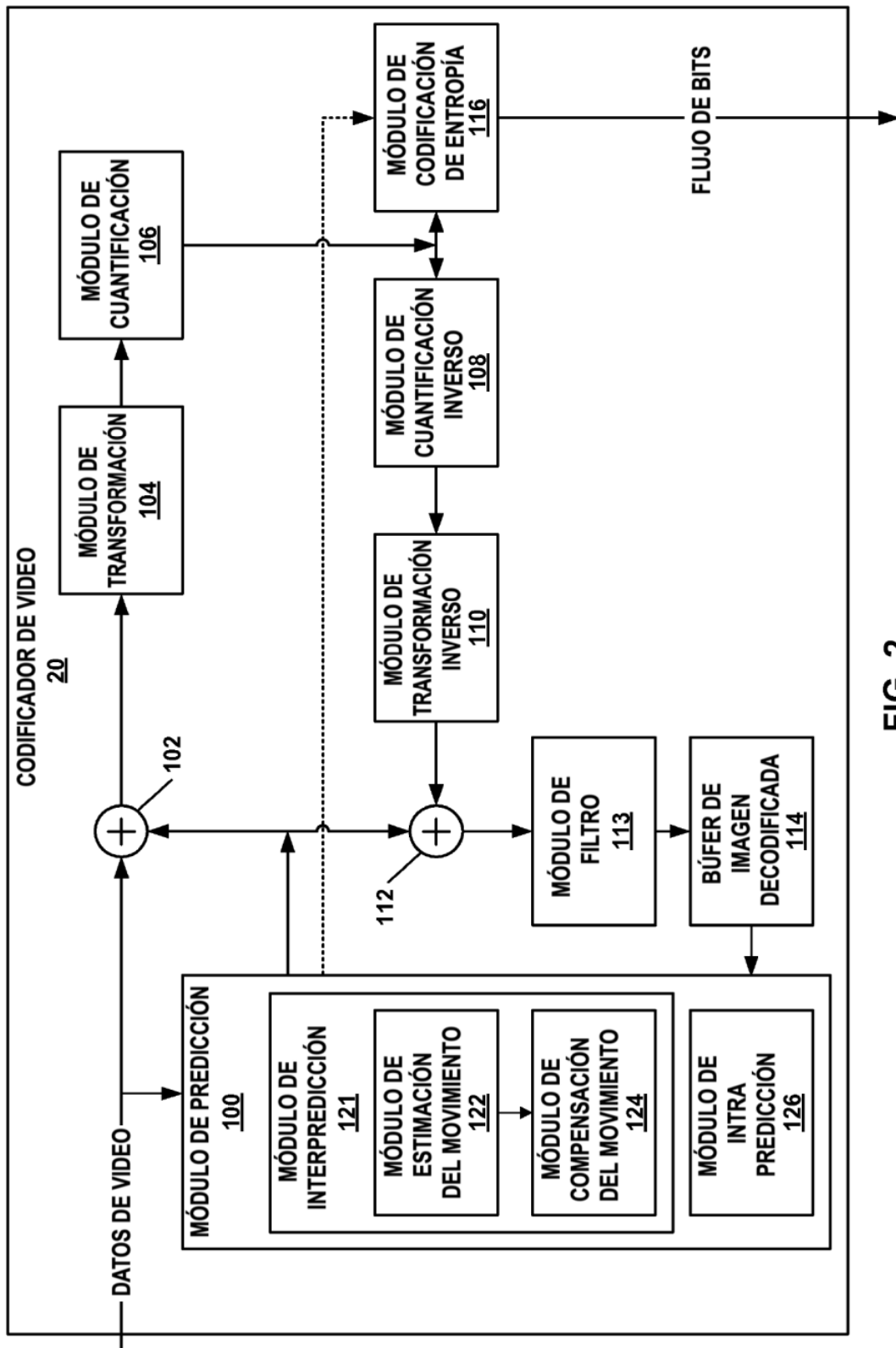


FIG. 1



**FIG. 2**



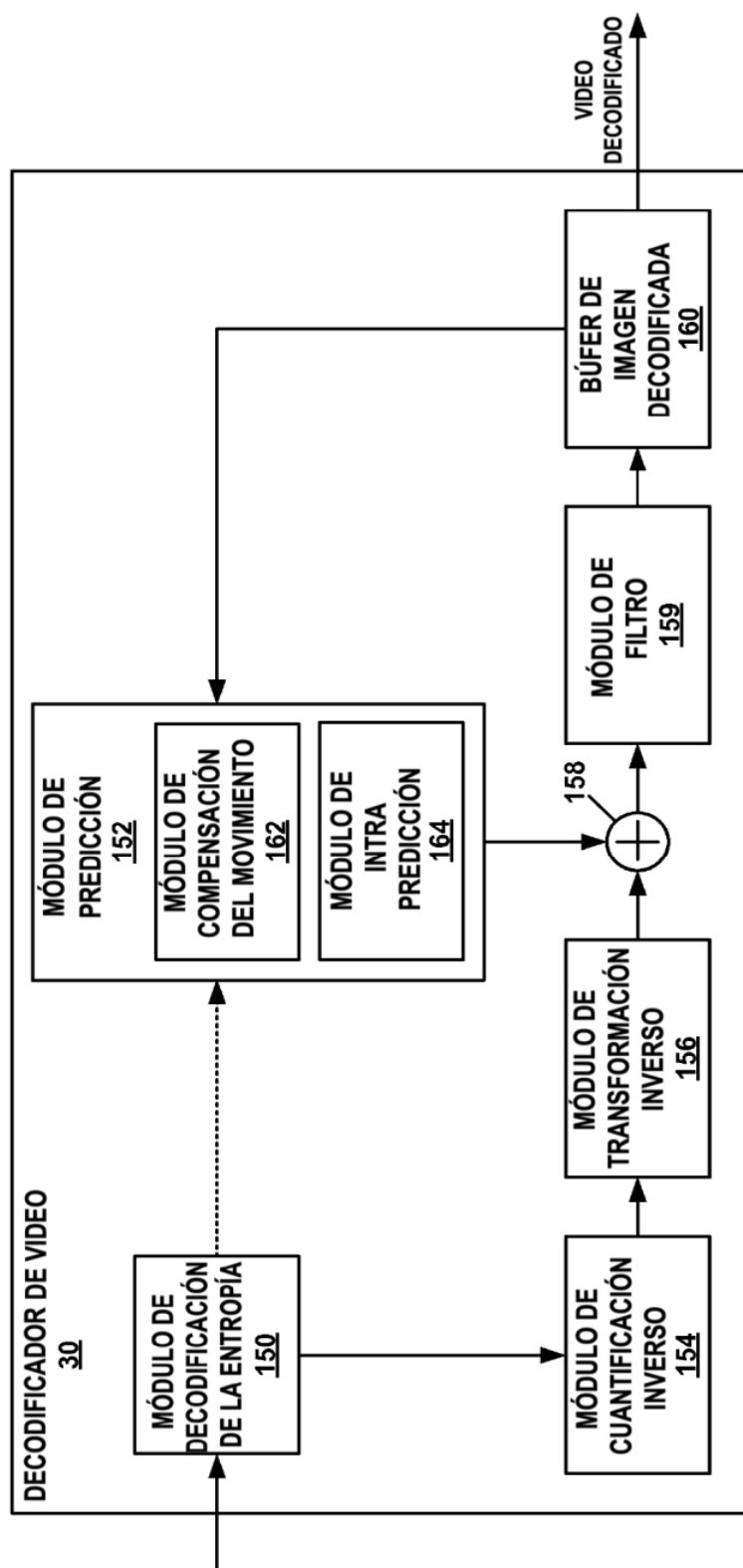


FIG. 3

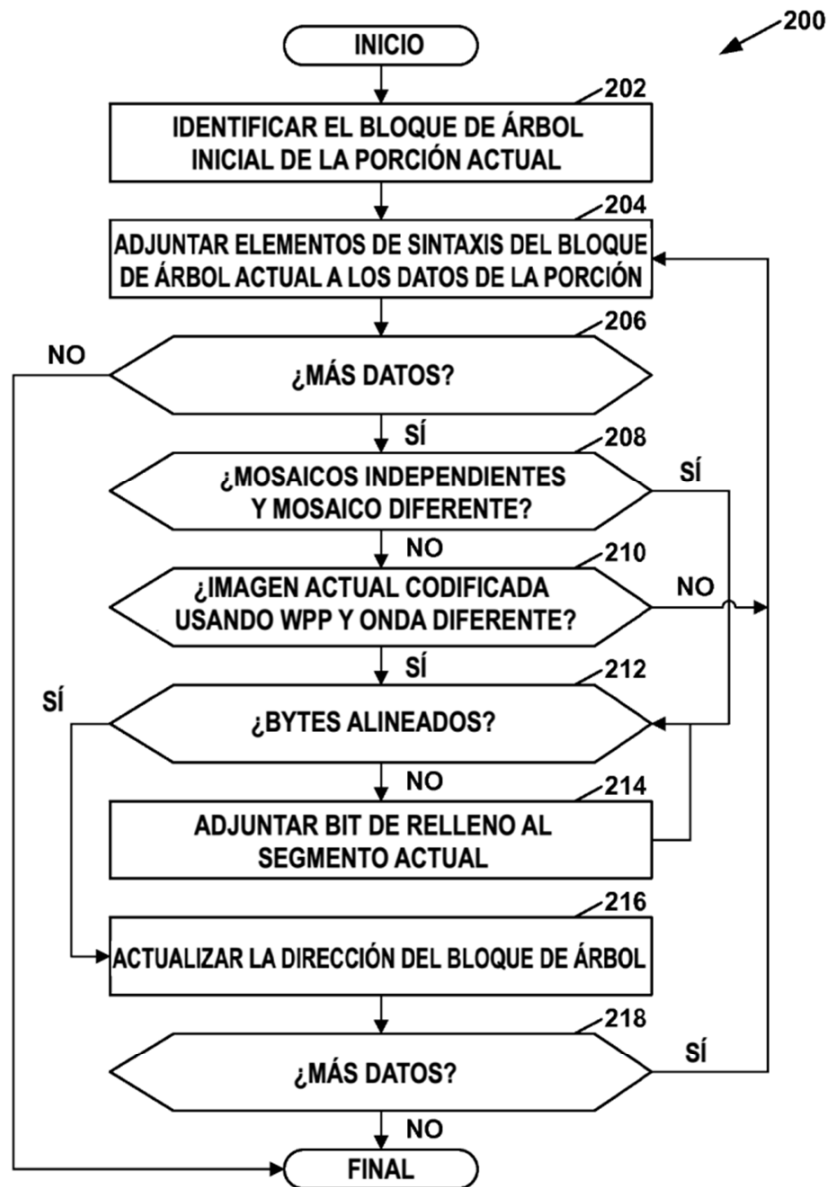
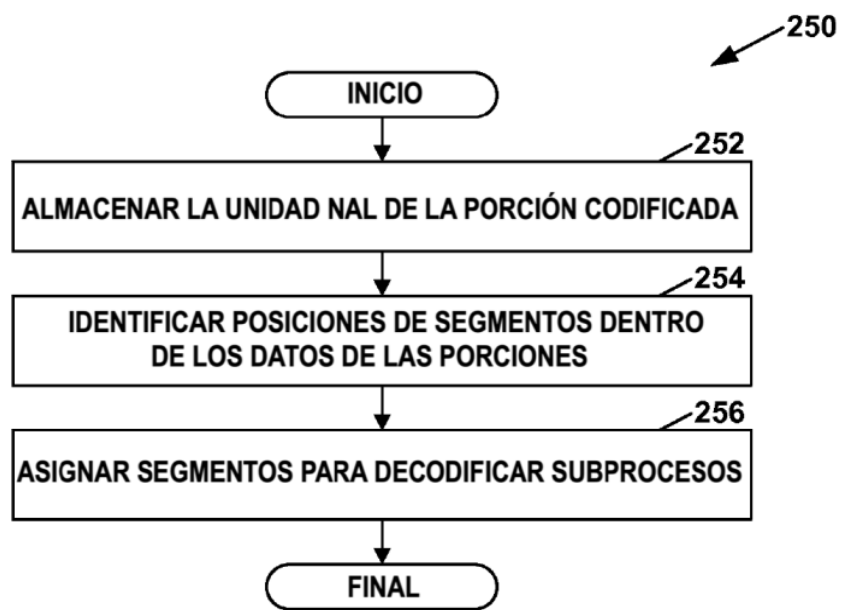


FIG. 4



**FIG. 5**

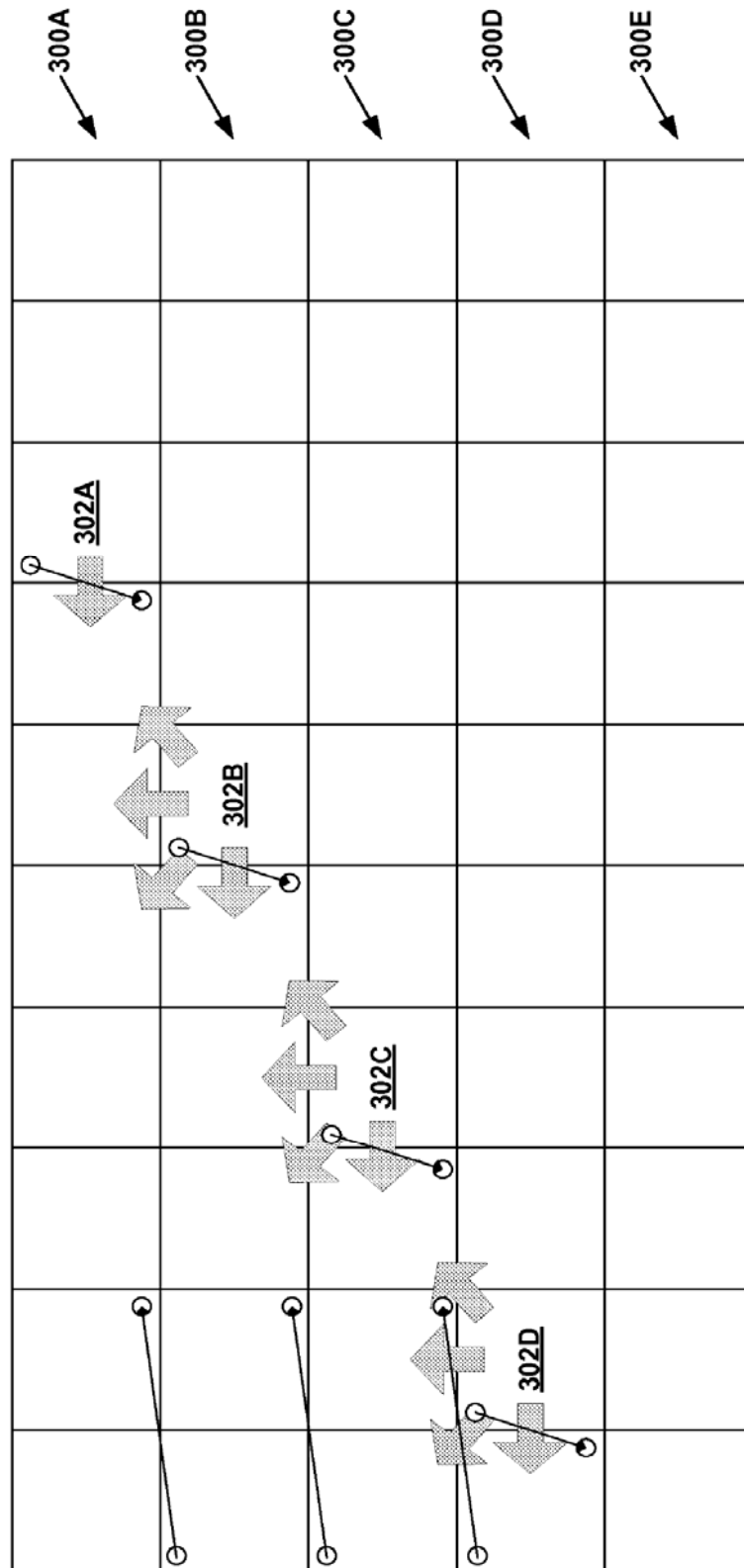


FIG. 6

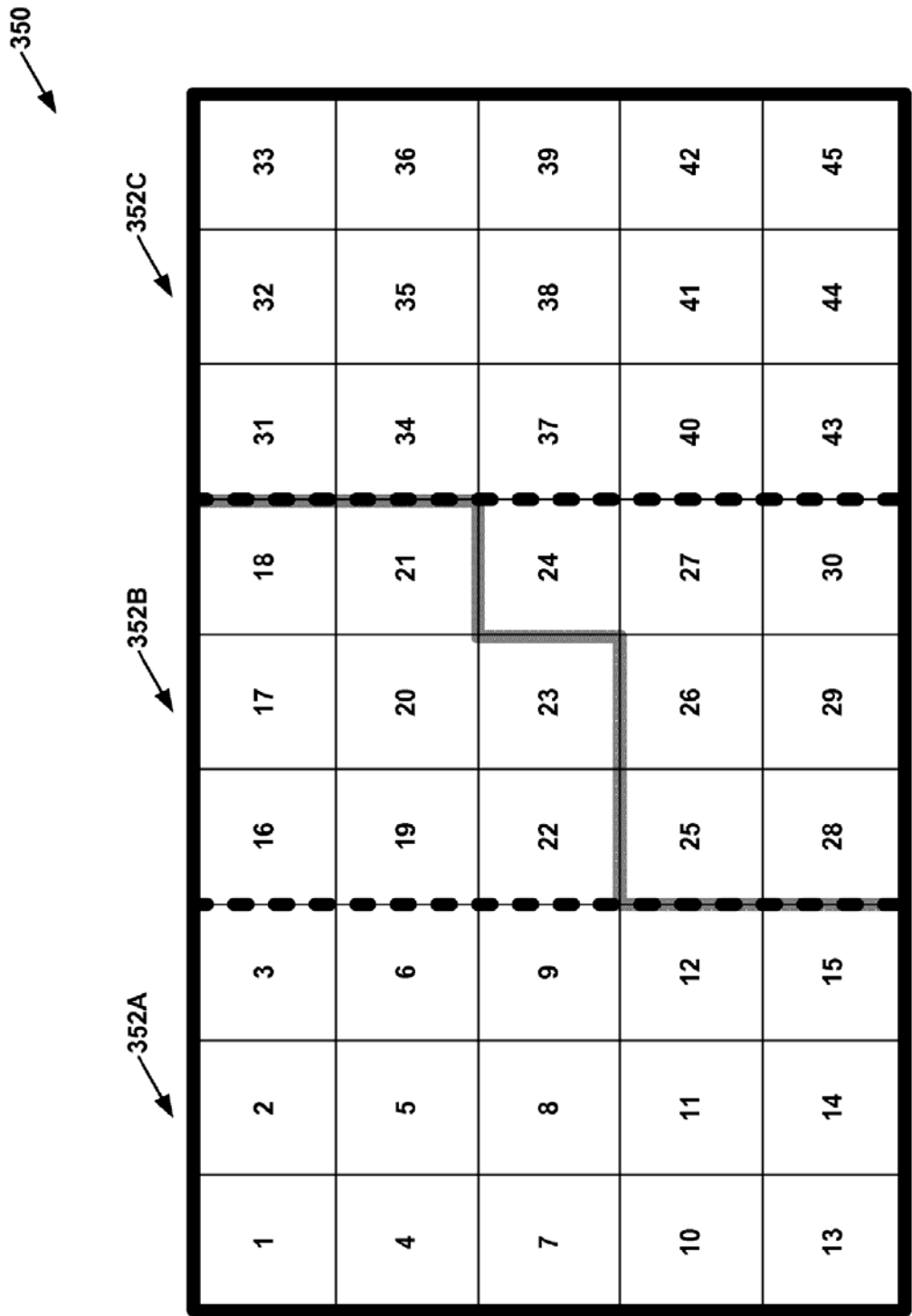


FIG. 7

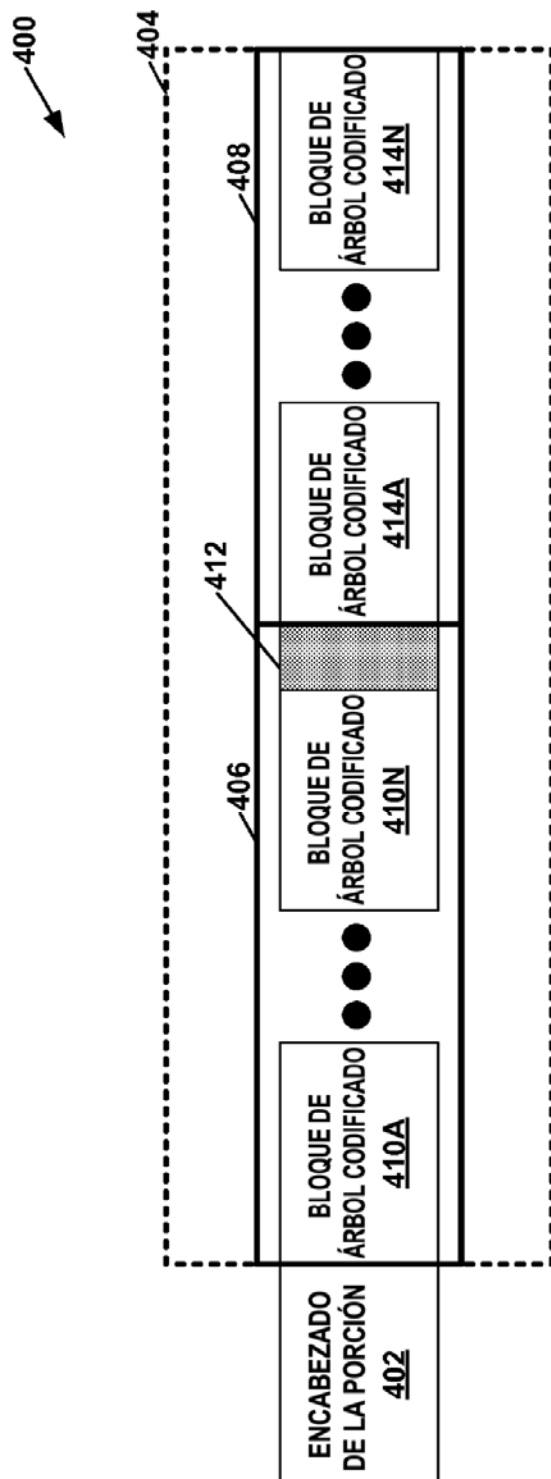


FIG. 8