(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 892 341**

(51) Int. CI.:

H04N 19/13 (2014.01)
H04N 19/17 (2014.01)
H04N 19/174 (2014.01)
H04N 19/176 (2014.01)
H04N 19/436 (2014.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **02.11.2012 PCT/US2012/063265**

(87) Fecha y número de publicación internacional: **10.05.2013 WO13067311**

(96) Fecha de presentación y número de la solicitud europea: **02.11.2012 E 12787613 (4)**

(97) Fecha y número de publicación de la concesión europea: **08.09.2021 EP 2774366**

(54) Título: **Codificación de vídeo con unidades de capa de abstracción de red que incluyen múltiples particiones de imagen codificada**

(30) Prioridad:

04.11.2011 US 201161555932 P
08.11.2011 US 201161557259 P
01.11.2012 US 201213666442

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
03.02.2022

(73) Titular/es:

QUALCOMM INCORPORATED (100.0%)
International IP Administration, 5775 Morehouse Drive
San Diego, California 92121-1714, US

(72) Inventor/es:

WANG, YE-KUI

(74) Agente/Representante:

ISERN JARA, Jorge

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

Codificación de vídeo con unidades de capa de abstracción de red que incluyen múltiples particiones de imagen codificada

Esta solicitud reivindica el beneficio de la Solicitud de Patente Provisional de Estados Unidos N.° 61/555.932, presentada el 4 de noviembre de 2011, y de la Solicitud de Patente Provisional de Estados Unidos N.° 61/557.259, presentada el 8 de noviembre de 2011.

Campo técnico

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

Antecedentes

Las capacidades de vídeo digital se pueden incorporar en una amplia gama de dispositivos, incluidos televisores digitales, sistemas de transmisión digital directa, sistemas de transmisión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, tabletas, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, radioteléfonos móviles o por satélite, los llamados "teléfonos inteligentes", dispositivos de vídeo teleconferencia, dispositivos de transmisión de vídeo y similares. Los dispositivos de vídeo digital implementan técnicas de compresión de vídeo, como las descritas en los estándares definidos por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificación de Vídeo Avanzada (AVC), la norma de codificación de vídeo de alta eficiencia (HEVC) actualmente en desarrollo y extensiones de dichas normas. Los dispositivos de vídeo pueden transmitir, recibir, codificar, decodificar y/o almacenar información de vídeo digital de manera más eficiente implementando tales técnicas de compresión de vídeo.

Las técnicas de compresión de vídeo realizan predicción espacial (intra imagen) y/o predicción temporal (inter imagen) para reducir o eliminar la redundancia intrínseca en las secuencias de vídeo. Para la codificación de vídeo basada en bloques, un corte de vídeo (es decir, un fotograma de vídeo o una parte de un fotograma de vídeo) puede subdividirse en bloques de vídeo, que también pueden denominarse bloques de árbol, unidades de codificación (CU) y/o nodos de codificación. Los bloques de vídeo en un corte intra-codificado (I) de una imagen se codifican utilizando la predicción espacial con respecto a las muestras de referencia en los bloques vecinos de la misma imagen. Los bloques de vídeo en un corte inter-codificado (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 fotogramas y las imágenes de referencia pueden denominarse fotogramas de referencia.

La predicción espacial o temporal da como resultado un bloque predictivo para un bloque a codificar. Los datos residuales representan diferencias de píxel entre el bloque original a codificar y el bloque predictivo. Un bloque inter-codificado se codifica de acuerdo con un vector de movimiento que apunta a un bloque de muestras de referencia que forma el bloque predictivo, e indicando los datos residuales la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intra-codificado se codifica de acuerdo con un modo de intra-codificación y los datos residuales. Para una mayor compresión, los datos residuales pueden transformarse del dominio de píxeles a un dominio de transformada, lo que da como resultado coeficientes residuales, que a continuación pueden cuantificarse. Los coeficientes cuantificados, dispuestos inicialmente en una matriz bidimensional, pueden explorarse para producir un vector unidimensional de coeficientes, y se puede aplicar codificación por entropía para lograr una compresión aún mayor.

Sumario

La invención se expone en las reivindicaciones adjuntas.

Los detalles de uno o más ejemplos de la divulgación se establecen en los dibujos adjuntos y la descripción a continuación. Otras características, objetos y ventajas resultarán evidentes a partir de la descripción, los dibujos y las reivindicaciones.

Breve descripción de los dibujos

La Figura 1 es un diagrama de bloques que ilustra un sistema de codificación de vídeo de ejemplo que puede utilizar las técnicas descritas en esta divulgación.

La Figura 2 es un diagrama de bloques que ilustra un codificador de vídeo de ejemplo que puede implementar las técnicas descritas en esta divulgación.

La Figura 3 es un diagrama de bloques que ilustra un decodificador de vídeo de ejemplo que puede implementar las técnicas descritas en esta divulgación.

La Figura 4 es un diagrama de flujo que ilustra una operación de ejemplo de un codificador de vídeo, de acuerdo

con una o más técnicas de esta divulgación.

La Figura 5 es un diagrama de flujo que ilustra una operación de ejemplo de un decodificador de vídeo, de acuerdo con una o más técnicas de esta descripción.

La Figura 6 es un diagrama de flujo que ilustra una operación de ejemplo del codificador de vídeo para generar una unidad de NAL de corte codificada que tiene un encabezado de corte alineado en bytes, de acuerdo con una o más técnicas de esta descripción.

La Figura 7 es un diagrama de flujo que ilustra una operación de ejemplo del codificador de vídeo para generar un encabezado de corte alineado en bytes, de acuerdo con una o más técnicas de esta descripción.

La Figura 8 es un diagrama de flujo que ilustra una operación de ejemplo del decodificador de vídeo en el que el decodificador de vídeo recibe una unidad de NAL de corte codificada que tiene un encabezado de corte alineado en bytes, de acuerdo con una o más técnicas de esta descripción.

La Figura 9 es un diagrama conceptual que ilustra el procesamiento paralelo de frente de onda.

La Figura 10 es un diagrama conceptual que ilustra un orden de codificación de ejemplo cuando se subdivide una imagen en una pluralidad de piezas.

La Figura 11 es un diagrama de flujo que ilustra una operación de ejemplo del codificador de vídeo para generar un flujo de bits que incluye valores de longitud variable que representan elementos de sintaxis asociados con desplazamientos de particiones de imagen codificada dentro de una unidad de NAL, de acuerdo con uno o más aspectos de esta divulgación.

La Figura 12 es un diagrama de flujo que ilustra una operación de ejemplo del decodificador de vídeo para decodificar datos de vídeo representados por un flujo de bits que incluye valores de longitud variable que representan elementos de sintaxis asociados con desplazamientos de particiones de imagen codificada dentro de una unidad de NAL, de acuerdo con uno o más aspectos de esta divulgación.

Descripción detallada

Una imagen incluye una pluralidad de bloques de árbol. Los bloques de árbol están asociados con bloques de píxeles bidimensionales dentro de la imagen. Un codificador de vídeo divide la imagen en una pluralidad de particiones de imagen. Por ejemplo, el codificador de vídeo puede dividir la imagen en cortes de entropía, piezas u ondas de procesamiento paralelo de frente de onda (WPP). En otras palabras, esta divulgación puede utilizar la expresión "partición de imagen" para referirse genéricamente a cortes de entropía, piezas u ondas WPP. Las particiones de la 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 exactamente con una de las particiones de la imagen. Un codificador de vídeo puede generar una unidad de capa de abstracción de red (NAL) de corte codificado que incluye una o más particiones de imagen codificada asociadas con un corte.

En algunos ejemplos, un decodificador de vídeo puede decodificar particiones de imagen codificada de un corte en paralelo. Para decodificar las particiones de imagen codificada de un corte en paralelo, el codificador de vídeo puede señalizar, dentro de una unidad de NAL de corte codificada, los puntos de entrada de las particiones de imagen codificada. Un punto de entrada de una partición de imagen codificada puede ser la posición de un primer bit de la partición de imagen codificada. Debido a que el decodificador de vídeo puede determinar los puntos de entrada de las particiones de imagen codificada, el decodificador de vídeo puede asignar diferentes particiones de imagen codificada a diferentes núcleos de procesamiento de un decodificador para decodificación en paralelo.

Para señalizar los puntos de entrada de las piezas, el codificador de vídeo puede generar una serie de elementos de sintaxis de desplazamiento. Cada uno de los elementos de sintaxis de desplazamiento puede indicar el número de bytes en una pieza. Además, el codificador de vídeo puede generar dos elementos de sintaxis adicionales. El número de piezas en un corte se puede determinar basándose en el primer elemento de sintaxis adicional. El número de bits utilizados en cada uno de los elementos de sintaxis de desplazamiento se puede determinar basándose en el segundo elemento de sintaxis adicional. En las implementaciones existentes, cada uno de los elementos de sintaxis adicionales siempre se representa en el flujo de bits utilizando un número entero sin signo de 5 bits.

Representar siempre los elementos de sintaxis adicionales como números enteros de 5 bits sin signo puede aumentar innecesariamente el número de bits en el flujo de bits. Además, representar siempre los elementos de sintaxis adicionales como números enteros sin signo de 5 bits puede limitar el número de elementos de sintaxis de desplazamiento a 32. De acuerdo con una o más técnicas de esta divulgación, un codificador de vídeo puede codificar por entropía (por ejemplo, usar ue (v), elemento de sintaxis codificado por Exp-Golomb de orden 0 de número entero sin signo con el bit izquierdo primero) los elementos de sintaxis adicionales. Debido a que la entropía del codificador de vídeo codifica los elementos de sintaxis adicionales, los elementos de sintaxis adicionales se representan en el flujo de bits utilizando secuencias de bits de longitud variable. En muchos casos, estas secuencias de bits de longitud variable pueden representar los elementos de sintaxis adicionales utilizando menos bits que los números enteros de 5 bits sin signo. Además, la codificación por entropía de estos elementos de sintaxis adicionales puede permitir que el codificador de vídeo señalice más de 32 elementos de sintaxis de desplazamiento.

En algunos casos, es posible que un decodificador de vídeo en particular no esté configurado para decodificar particiones de imagen codificada en paralelo. Es decir, algunos decodificadores de vídeo pueden tener diferentes capacidades en términos de procesamiento paralelo. En tales casos, los bits usados para señalar los puntos de entrada

de las particiones de imagen codificada no son útiles para el decodificador de vídeo.

De acuerdo con las técnicas de esta divulgación, un codificador de vídeo puede generar una unidad de NAL de VCL que contiene una pluralidad de particiones de imagen codificada de los datos de vídeo. Además, el codificador de vídeo puede generar un mensaje de información de mejora complementaria (SEI) que indica los puntos de entrada de las particiones de imagen codificada dentro de la unidad de NAL. Para facilitar la explicación, esta descripción puede hacer referencia a un mensaje de SEI que indica puntos de entrada de particiones de imagen codificada en una unidad de NAL de corte codificada como un "mensaje de SEI de punto de entrada". El mensaje de SEI de punto de entrada puede proporcionarse, por ejemplo, en una unidad de NAL no de VCL, tal como una unidad de NAL no de VCL que tiene un tipo SEI.

Los elementos de red con reconocimiento de medios (MANE) pueden configurarse con datos que indican las configuraciones de decodificadores de vídeo particulares. Basándose en tales datos, un MANE puede configurarse para transmitir un mensaje de SEI de punto de entrada a un decodificador de vídeo solo si el decodificador de vídeo está configurado para decodificar particiones de imagen codificada en paralelo. Debido a que MANE no transmite datos que el decodificador de vídeo no está configurado para usar, es posible que se conserve el ancho de banda. Además, debido a que los puntos de entrada se señalizan en mensajes de SEI, puede que no sea necesario señalizar en una unidad de NAL de corte codificada si la unidad de NAL de corte codificada incluye o no datos que indican posiciones de particiones de imagen codificada. Esto ayuda a reducir el número de bits en la unidad de NAL de corte codificada y, por lo tanto, conserva aún más el ancho de banda.

Los dibujos adjuntos ilustran ejemplos. Elementos indicados por números de referencia en los dibujos adjuntos corresponden a elementos indicados por números de referencia similares en la siguiente descripción. En esta divulgación, elementos que tienen nombres que comienzan con palabras ordinales (por ejemplo, "primero", "segundo", "tercero" y así sucesivamente) no implican necesariamente que los elementos tienen un orden particular. En su lugar, tales palabras ordinales se usan solamente para referirse a diferentes elementos de un mismo o similar tipo.

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

Como se muestra en la Figura 1, el sistema 10 de codificación de vídeo incluye un dispositivo 12 de origen y un dispositivo 14 de destino. El dispositivo 12 de origen genera datos de vídeo codificados. Por consiguiente, el dispositivo 12 de origen puede denominarse como un dispositivo de codificación de vídeo o aparato de codificación de vídeo. El dispositivo 14 de destino puede decodificar los datos de vídeo codificados generados por el dispositivo 12 de origen. Por consiguiente, el dispositivo 14 de destino puede denominarse dispositivo de decodificación de vídeo o aparato de decodificación de vídeo. El dispositivo 12 de origen y el dispositivo 14 de destino pueden ser ejemplos de dispositivos de codificación de vídeo o aparatos de codificación de vídeo.

El dispositivo 12 de origen y el dispositivo 14 de destino pueden comprender una amplia gama de dispositivos, incluidos ordenadores de sobremesa, dispositivos informáticos móviles, ordenadores portátiles (por ejemplo, portátiles), ordenadores de tableta, decodificadores de salón, teléfonos móviles como los llamados teléfonos "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, ordenadores para el automóvil o similares.

El dispositivo 14 de destino puede recibir datos de vídeo codificados desde el dispositivo 12 de origen a través de un canal 16. El canal 16 puede comprender un tipo de medio o dispositivo capaz de mover los datos de vídeo codificados desde el dispositivo 12 de origen al dispositivo 14 de destino. En un ejemplo, el canal 16 puede comprender uno o más medios de comunicación que permiten que el dispositivo 12 de origen transmita datos de vídeo codificados directamente al dispositivo 14 de destino en tiempo real. En este ejemplo, el dispositivo 12 de origen puede modular los datos de vídeo codificados de acuerdo con una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y puede transmitir los datos de vídeo modulados al dispositivo 14 de destino. El uno o más medios de comunicación pueden incluir medios de comunicación inalámbricos y/o alámbricos, tales como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión físicas. El uno o más medios de comunicación pueden formar parte de una red basada en paquetes, tal como una red de área local, una red de área amplia o una red global (por ejemplo, Internet). El uno o más medios de comunicación pueden incluir encaminadores, conmutadores, estaciones base u otro equipo que facilite la comunicación desde el dispositivo 12 de origen al dispositivo 14 de destino.

En otro ejemplo, el canal 16 puede incluir un medio de almacenamiento que almacena datos de vídeo codificados generados por el dispositivo 12 de origen. En este ejemplo, el dispositivo 14 de destino puede acceder al medio de almacenamiento mediante acceso al disco o acceso a la tarjeta. El medio de almacenamiento puede incluir una diversidad de medios de almacenamiento de datos a los que se accede localmente, tales como discos Blu-ray, DVD, CD-ROM, memoria flash u otros medios de almacenamiento digital adecuados para almacenar datos de vídeo codificados.

En un ejemplo adicional, el canal 16 puede incluir un servidor de archivos u otro dispositivo de almacenamiento intermedio que almacena vídeo codificado generado por el dispositivo 12 de origen. En este ejemplo, el dispositivo 14 de destino puede acceder a datos de vídeo codificados almacenados en el servidor de archivos u otro dispositivo de almacenamiento intermedio mediante envío por flujo continuo o descarga. El servidor de archivos puede ser un tipo de servidor capaz de almacenar datos de vídeo codificados y transmitir los datos de vídeo codificados al dispositivo 14 de destino. Los servidores de archivos de ejemplo incluyen servidores web (por ejemplo, para un sitio web), servidores de protocolo de transferencia de archivos (FTP), dispositivos de almacenamiento conectados a la red (NAS) y unidades de disco locales.

El dispositivo 14 de destino puede acceder a los datos de vídeo codificados a través de una conexión de datos convencional, como una conexión a Internet. Ejemplos de tipos de conexiones de datos pueden incluir canales inalámbricos (por ejemplo, conexiones Wi-Fi), conexiones por cable (por ejemplo, DSL, módem por cable, etc.) o combinaciones de ambos que son adecuadas para acceder a datos de vídeo codificados almacenados en un servidor de archivos. La transmisión de datos de vídeo codificados desde el servidor de archivos puede ser una transmisión por envío de flujo continuo, 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 vídeo para soportar una diversidad de aplicaciones multimedia, tales como transmisiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo de envío por flujo continuo, por ejemplo, a través de Internet, codificación de datos de vídeo para almacenamiento en un medio de almacenamiento de datos, decodificación de datos de vídeo almacenados en un medio de almacenamiento de datos u otras aplicaciones. En algunos ejemplos, el sistema 10 de codificación de vídeo puede configurarse para soportar una transmisión de vídeo unidireccional o bidireccional para soportar aplicaciones tales como difusión en continuo de vídeo, reproducción de vídeo, difusión de vídeo y/o telefonía de vídeo.

En el ejemplo de la Figura 1, el dispositivo 12 de origen incluye un origen 18 de vídeo, un codificador 20 de vídeo y una interfaz 22 de salida. En algunos ejemplos, la interfaz 22 de salida puede incluir un modulador/demodulador (módem) y/o un transmisor. El origen 18 de vídeo puede incluir un dispositivo de captura de vídeo, por ejemplo, una cámara de vídeo, un archivo de vídeo que contiene datos de vídeo capturados previamente, una interfaz de alimentación de vídeo para recibir datos de vídeo de un proveedor de contenido de vídeo y/o un sistema de gráficos por ordenador para generar datos de vídeo o una combinación de tales orígenes de datos de vídeo.

El codificador 20 de vídeo puede codificar datos de vídeo del origen 18 de vídeo. En algunos ejemplos, el dispositivo 12 de origen puede transmitir directamente los datos de vídeo codificados al dispositivo 14 de destino a través de la interfaz 22 de salida. Los datos de vídeo codificados también pueden almacenarse en un medio de almacenamiento o un servidor de archivos para un acceso posterior por el dispositivo 14 de destino para decodificación y/o reproducción.

En el ejemplo de la Figura 1, el dispositivo 14 de destino incluye una interfaz 28 de entrada, un decodificador 30 de vídeo y un dispositivo 32 de visualización. En algunos ejemplos, la interfaz 28 de entrada puede incluir un receptor y/o un módem. La interfaz 28 de entrada del dispositivo 14 de destino puede recibir datos de vídeo codificados por el canal 16. El dispositivo 32 de visualización puede estar integrado o puede ser externo al dispositivo 14 de destino. En general, el dispositivo 32 de visualización muestra datos de vídeo decodificados. El dispositivo 32 de visualización puede comprender una diversidad de dispositivos de visualización, tales como una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodo emisor de luz orgánica (OLED) u otro tipo de dispositivo de visualización.

El codificador 20 de vídeo y el decodificador 30 de vídeo pueden funcionar de acuerdo con una norma de compresión de vídeo, como la norma de codificación de vídeo de alta eficiencia (HEVC) actualmente en desarrollo, y pueden ajustarse a un modelo de prueba de HEVC (HM). En Bross se describe un borrador reciente de la próxima norma HEVC, denominado "HEVC Working Draft 4" o "WD4". *et al.*, "WD4: Working Draft 4 of High-Efficiency Video Coding", Joint Collaborative Team on Video Coding (JCT-VC) de la ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 6ª reunión: Torino, Italia, julio de 2011, que, al 27 de septiembre de 2012, se puede descargar de: http://phenix.int-evry.fr/jct/doc_end_user/documents/6_Torino/wg11/JCTVC-F803-v3.zip.

Como alternativa, el codificador 20 de vídeo y el decodificador 30 de vídeo pueden operar de acuerdo con otros estándares de propiedad o de la industria, incluidos ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 o ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual e ITU-T H.264 (también conocido como ISO/IEC MPEG-4 AVC), incluidas sus extensiones Scalable vídeo Coding (SVC) y Multiview vídeo Coding (MVC). Sin embargo, las técnicas de esta divulgación no se limitan a ninguna norma o técnica de codificación particular.

De nuevo, la Figura 1 es simplemente un ejemplo y las técnicas de esta divulgación pueden aplicarse a configuraciones de codificación de vídeo (por ejemplo, codificación de vídeo o decodificación de vídeo) que no incluyen necesariamente ninguna comunicación de datos entre los dispositivos de codificación y decodificación. En otros ejemplos, los datos pueden recuperarse de una memoria local, enviarse por flujo continuo a través de una red o similares. 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 se realiza mediante 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 vídeo y el decodificador 30 de vídeo pueden implementarse cada uno como cualquiera de una diversidad de circuitos adecuados, como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de aplicación (ASIC), campos de matrices de puertas programables (FPGA), lógica discreta, hardware o cualquier combinación de los mismos. Si las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio de almacenamiento adecuado, no transitorio, legible por ordenador y puede ejecutar las instrucciones en hardware usando uno o más procesadores para realizar las técnicas de esta divulgación. Cualquiera de los anteriores (incluido hardware, software, una combinación de hardware y software, etc.) puede considerarse uno o más procesadores. Cada uno del codificador 20 de vídeo y del decodificador 30 de vídeo se puede incluir 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.

Esta divulgación puede referirse en general al codificador 20 de vídeo que "señaliza" cierta información a otro dispositivo, tal como el decodificador 30 de vídeo. Debe entenderse, sin embargo, que el codificador 20 de vídeo puede señalizar información asociando ciertos elementos de sintaxis con diversas porciones codificadas de datos de vídeo. Es decir, el codificador 20 de vídeo puede "señalizar" datos almacenando ciertos elementos de sintaxis en diversas porciones codificadas de datos de vídeo. En algunos casos, tales elementos de sintaxis pueden codificarse y almacenarse (por ejemplo, en un sistema de almacenamiento) antes de recibirse y decodificarse por el decodificador 30 de vídeo. Por tanto, el término "señalización" puede hacer referencia en general a la comunicación de elementos de sintaxis y/u otros datos utilizados para decodificar los datos de vídeo comprimidos. Tal comunicación puede tener lugar en tiempo real o casi real. Como alternativa, tal comunicación puede ocurrir durante un lapso de tiempo, como podría ocurrir al almacenar elementos de sintaxis en un medio de almacenamiento legible por ordenador en un flujo de bits codificado en el momento de la codificación, que a continuación puede recuperarse por un dispositivo de decodificación en cualquier momento después de almacenarse en este medio.

Como se mencionó brevemente anteriormente, el codificador 20 de vídeo codifica datos de vídeo. Los datos de vídeo pueden comprender una o más imágenes. Cada una de las imágenes puede ser una imagen fija que forma parte de un vídeo. En algunos casos, una imagen puede denominarse como un "fotograma" de vídeo. El codificador 20 de vídeo puede generar un flujo de bits que incluye una secuencia de bits que forman una representación codificada de los datos de vídeo. 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 vídeo puede generar una serie de imágenes codificadas y datos asociados. Las imágenes codificadas pueden ser representaciones codificadas de imágenes en los datos de vídeo. Los datos asociados pueden incluir conjuntos de parámetros de secuencia (SPS), conjuntos de parámetros de imagen (PPS), conjuntos de parámetros de adaptación (APS) y otras estructuras de sintaxis. Un SPS puede contener parámetros aplicables a cero o más secuencias de imágenes. Un PPS puede contener parámetros aplicables a cero o más imágenes. Un APS puede contener parámetros aplicables a cero o más imágenes. Los parámetros en un APS pueden ser parámetros que es más probable que cambien que parámetros en un PPS.

Para generar una representación codificada de una imagen, el codificador 20 de vídeo puede dividir la imagen en una pluralidad de bloques 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 normas anteriores, tal 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).

Cada uno de los bloques de árbol puede estar asociado con un bloque de píxeles de igual tamaño diferente dentro de la imagen. Cada píxel puede comprender una muestra de luminancia (Y), una muestra de crominancia Cb y una muestra de crominancia Cr. Por tanto, cada bloque de árbol puede asociarse con un bloque de muestras de luminancia (Y), un bloque de muestras de crominancia Cb de la imagen y un bloque de muestras de crominancia Cr de la imagen. Para facilitar la explicación, esta descripción puede referirse a una matriz bidimensional de píxeles como un bloque de píxeles y puede referirse a una matriz bidimensional de muestras como un bloque de muestra. El codificador 20 de vídeo puede utilizar la subdivisión en árbol cuádruple para subdividir los bloques de píxeles asociados con un bloque de árbol en bloques de píxeles asociados con las CU, de ahí el nombre "bloques de árbol".

El codificador 20 de vídeo puede dividir una imagen en una pluralidad de cortes. Cada uno de los cortes puede incluir un número entero de bloques de árbol. Como parte de la codificación de una imagen, el codificador 20 de vídeo puede generar representaciones codificadas de cada corte de la imagen, es decir, cortes codificados. Para generar un corte codificado, el codificador 20 de vídeo puede codificar cada bloque de árbol del corte para generar representaciones codificadas de cada uno de los bloques de árbol del corte (es decir, bloques de árbol codificados).

Para generar un bloque de árbol codificado, el codificador 20 de vídeo puede realizar de forma recursiva una subdivisión de árbol cuádruple en el bloque de píxeles asociado con un bloque de árbol para dividir el bloque de píxeles

en bloques de píxeles progresivamente más pequeños. Cada uno de los bloques de píxeles más pequeños puede estar asociado con una CU. Una CU particionada puede ser una CU cuyo bloque de píxeles está subdividido en bloques de píxeles asociados con otras CU. Una CU no subdividida puede ser una CU cuyo bloque de píxeles no está dividido en bloques de píxeles asociados con otras CU.

El codificador 20 de vídeo puede generar una o más unidades de predicción (PU) para cada CU no subdividida. Cada una de las PU de una CU puede estar asociada con un bloque de píxeles diferente dentro del bloque de píxeles de la CU. El codificador 20 de vídeo puede generar bloques de píxeles predictivos para cada PU de la CU. Los bloques de píxeles predictivos de una PU pueden ser bloques de píxeles.

El codificador 20 de vídeo puede utilizar intra predicción o inter predicción para generar el bloque de píxeles predictivo para una PU. Si el codificador 20 de vídeo usa intra predicción para generar el bloque de píxeles predictivo de una PU, el codificador 20 de vídeo puede generar el bloque de píxeles predictivo de la PU basándose en píxeles decodificados de la imagen asociada con la PU. Si el codificador 20 de vídeo usa la inter predicción para generar el bloque de píxeles predictivo de la PU, el codificador 20 de vídeo puede generar el bloque de píxeles predictivo de la PU basándose en píxeles decodificados de una o más imágenes distintas de la imagen asociada con la PU.

El codificador 20 de vídeo puede generar un bloque de píxeles residual para una CU basándose en bloques de píxeles predictivos de las PU de la CU. El bloque de píxeles residuales para la CU puede indicar diferencias entre las muestras en los bloques de píxeles predictivos para las PU de la CU y las muestras correspondientes en los bloques de píxeles originales de la CU.

Además, como parte de la codificación de una CU no particionada, el codificador 20 de vídeo puede realizar subdivisiones en árbol cuádruple recursivas en los bloques de píxeles residuales de la CU para dividir los bloques de píxeles residuales de la CU en uno o más bloques de píxeles residuales más pequeños asociados con las unidades de transformada (TU) de la CU. De esta manera, cada TU de la CU puede asociarse con un bloque de muestra residual de muestras de luminancia (Y) y dos bloques de muestra residual de muestras de crominancia (Cr y Cb).

El codificador 20 de vídeo puede aplicar una o más transformadas a bloques de muestra residuales asociados con las TU para generar bloques de coeficiente (es decir, bloques de coeficientes) asociados con las TU. Conceptualmente, un bloque de coeficientes puede ser una matriz bidimensional de coeficientes. El codificador 20 de vídeo puede realizar un proceso de cuantificación en un bloque de coeficientes. La cuantificación generalmente se refiere a un proceso en el que los coeficientes se cuantifican para posiblemente reducir la cantidad de datos utilizados para representar los coeficientes, proporcionando una mayor compresión.

El codificador 20 de vídeo puede generar conjuntos de elementos de sintaxis que representan los coeficientes en el bloque de coeficientes cuantificados. El codificador 20 de vídeo puede aplicar operaciones de codificación por entropía, tales como operaciones de codificación aritmética binaria adaptativa al contexto (CABAC), a al menos algunos de estos elementos de sintaxis. Como parte de la realización de una operación de codificación por entropía, el codificador 20 de vídeo puede seleccionar un contexto de codificación. En el caso de CABAC, el contexto de codificación puede indicar probabilidades de binarios con valor 0 y con valor 1. En el caso de la codificación de longitud variable adaptativa al contexto (CAVLC), el contexto de codificación puede ser un conjunto de códigos de longitud variable. El codificador 20 de vídeo puede utilizar el contexto de codificación para codificar uno o más elementos de sintaxis.

El flujo de bits generado por el codificador 20 de vídeo puede incluir una serie de unidades de capa de abstracción de red (NAL). Cada una de las unidades de NAL puede ser una estructura de sintaxis que contiene una indicación de un tipo de datos en la unidad de NAL y bytes que contienen los datos. Por ejemplo, una unidad de NAL puede contener datos que representan un SPS, un PPS, un corte codificado, información de mejora complementaria (SEI), un delimitador de la unidad de acceso, datos de relleno u otro tipo de datos. Los datos en una unidad de NAL pueden incluir diversas estructuras de sintaxis. Las unidades de NAL de corte codificadas son unidades de NAL que incluyen cortes codificados. "Unidad de NAL de VCL" es un término colectivo para unidades de NAL de corte codificadas y el subconjunto de unidades de NAL que tienen valores reservados de nal_unit_type que se clasifican como unidades de NAL de VCL.

Las unidades de NAL de SEI son unidades de NAL que incluyen mensajes de SEI. Los mensajes de SEI incluyen información que puede ser útil para la resistencia a errores, el tiempo, la salida y así sucesivamente. Sin embargo, los mensajes de SEI no son necesarios en el proceso de decodificación. En otras palabras, un decodificador de vídeo puede decodificar datos de vídeo sin usar la información en los mensajes de SEI. Por lo tanto, en el contexto de una norma de codificación de vídeo tal como H.264/AVC o HEVC, los mensajes de SEI son normativos (es decir, no informativos) y son opcionales en el sentido de que los decodificadores de vídeo conformes pueden elegir simplemente descartar cualquier mensaje de SEI presente en un flujo de bits. Además, puede que no sea necesario transmitir mensajes de SEI asociados con datos de vídeo para que un decodificador de vídeo pueda decodificar los datos de vídeo.

El codificador 20 de vídeo puede dividir una imagen 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. El codificador 20

de vídeo puede dividir la imagen en una pluralidad de particiones de imagen de diversas formas. Como se describe a continuación, el codificador 20 de vídeo puede dividir la imagen en una pluralidad de cortes de entropía, una pluralidad de piezas o una pluralidad de ondas de procesamiento paralelo de frente de onda (WPP). Esta descripción puede utilizar la expresión "partición de imágenes" para referirse genéricamente a cortes de entropía, piezas y ondas WPP.

Como se mencionó anteriormente, el codificador 20 de vídeo puede dividir la imagen en una pluralidad de cortes de entropía. Los cortes de entropía también se pueden denominar cortes ligeros, cortes con un encabezado de corte corto o cortes cortos. Un corte de entropía puede incluir un subconjunto de las CU de un corte normal. En algunos ejemplos, el codificador 20 de vídeo puede subdividir las CU entre cortes de entropía de modo que ninguno de los cortes de entropía incluya más binarios (por ejemplo, bits codificados por entropía) que un límite superior. En algunos casos, una unidad de NAL de un solo corte codificado puede incluir múltiples cortes de entropía de un corte. En otros ejemplos, cada corte de entropía puede incluirse en una unidad de NAL separada.

Se permite la predicción dentro de la imagen a través de los límites de corte de entropía. Por ejemplo, si un codificador de vídeo está realizando una intra predicción en una PU particular, el codificador de vídeo puede usar muestras de una PU vecina, incluso si la PU vecina está en un corte de entropía diferente a la PU particular. En este ejemplo, es posible que el codificador de vídeo no pueda utilizar muestras de la PU vecina si la PU vecina está en un corte diferente al de la PU particular.

Sin embargo, cuando un codificador de vídeo está realizando codificación por entropía en datos asociados con una PU en particular, el codificador de vídeo está permitido únicamente a seleccionar contextos de codificación basados en información asociada con una PU vecina si la PU particular y la PU vecina están en el mismo corte de entropía. Debido a esta restricción, el codificador de vídeo puede realizar operaciones de codificación por entropía (es decir, codificación o decodificación por entropía) en múltiples cortes de entropía de un corte en paralelo. Sin embargo, el decodificador 30 de vídeo no puede reconstruir los bloques de píxeles de múltiples cortes de entropía de un corte en paralelo.

En algunos ejemplos, el codificador 20 de vídeo puede dividir una imagen en una o más piezas. Cada una de las piezas puede comprender un número entero de bloques de árbol en la imagen. El codificador 20 de vídeo puede dividir la imagen en piezas definiendo dos o más límites de piezas verticales y dos o más límites de piezas horizontales. Cada lado vertical de la imagen puede considerarse que es un límite de pieza vertical. Cada lado horizontal de la imagen actual puede considerarse que es un límite de pieza horizontal. Por ejemplo, si el codificador 20 de vídeo define cuatro límites de piezas verticales y tres límites de piezas horizontales para la imagen, la imagen actual se divide en seis piezas.

Un codificador de vídeo, tal como un codificador 20 de vídeo o un decodificador 30 de vídeo, puede codificar las piezas de una imagen de acuerdo con el orden de exploración por tramas. Además, el codificador de vídeo puede codificar cada bloque de árbol dentro de una pieza de acuerdo con un orden de exploración por tramas. De esta manera, el codificador de vídeo puede codificar cada bloque de árbol de una pieza dada de la imagen antes de codificar cualquier bloque de árbol de otra pieza de la imagen. En consecuencia, el orden en el que el codificador de vídeo codifica los bloques de árbol de la imagen puede ser diferente si la imagen se subdivide en múltiples piezas que si la imagen no se subdivide en múltiples piezas.

En algunos casos, un codificador de vídeo puede usar información asociada con una CU espacialmente vecina para realizar una intra predicción en una CU particular, siempre que la CU particular y la CU espacialmente vecina pertenezcan a la misma pieza y siempre que la CU particular y las CU espacialmente vecinas pertenezcan al mismo corte. La CU espacialmente vecina puede ser una CU que está asociada con un bloque de píxeles que es adyacente al bloque de píxeles asociado con una CU particular. En algunos casos, el codificador de vídeo puede utilizar información asociada con la CU espacialmente vecina para seleccionar un contexto para la codificación CABAC de un elemento de sintaxis de la CU particular, siempre que la CU particular y la CU espacialmente vecina estén dentro de la misma pieza. Debido a estas restricciones, el codificador de vídeo puede codificar en paralelo bloques de árbol de múltiples piezas.

Por ejemplo, la predicción dentro de la imagen, incluida la predicción del valor de píxel, la predicción del movimiento, la predicción del modo de codificación y la predicción del contexto de codificación por entropía, a través de los límites de las piezas, puede controlarse mediante una bandera tile_boundary_independence_idc. Si la bandera es igual a 1, no se permite la predicción dentro de la imagen a través de los límites de la pieza. De lo contrario, se permite la predicción dentro de la imagen a través de los límites de la pieza, excepto para los límites de la pieza que también son límites de la imagen o límites de corte. Si no se permite la predicción dentro de la imagen, la funcionalidad de las piezas puede ser cambiar el orden de exploración del bloque de árbol en comparación con el caso en el que no hay piezas, o, de manera equivalente, solo hay una pieza en una imagen. Si se permite la predicción dentro de las imágenes, además de cambiar el orden de exploración de bloques de árbol, las piezas también pueden proporcionar subdivisiones independientes que se pueden usar para codificación paralela (codificación y/o decodificación).

Independientemente de si se permite la predicción dentro de la imagen a través de los límites de las piezas, el codificador 20 de vídeo puede incluir múltiples piezas codificadas en un solo corte por entropía en una única unidad

de NAL. En este caso, incluso si se permite la predicción en imagen a través de los límites de pieza, no se permite la predicción del contexto de codificación por entropía a través de los límites de pieza. Además, en los ejemplos en los que una unidad de NAL puede contener solo un corte de entropía y un corte de entropía solo incluye las CU de una sola pieza, es posible que no sea necesario señalizar los puntos de entrada de cada pieza codificada del corte de entropía, ya que las piezas codificadas están en unidades de NAL separadas.

En otros ejemplos, el codificador de vídeo puede codificar una imagen usando procesamiento paralelo de frente de onda (WPP). Si el codificador de vídeo codifica la imagen usando WPP, el codificador de vídeo puede dividir los bloques de árbol de la imagen 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. Si el codificador de vídeo codifica la imagen usando WPP, el codificador de vídeo puede empezar a codificar una fila superior de bloques de árbol. Una vez que el codificador de vídeo ha codificado dos o más bloques de árbol de la fila superior, el codificador de vídeo puede empezar a codificar una fila de bloques de árbol de la segunda a la superior en paralelo con la codificación de la fila superior de bloques de árbol. Una vez que el codificador de vídeo ha codificado dos o más bloques de árbol de la segunda fila a la superior, el codificador de vídeo puede empezar a codificar una fila de la tercera a la superior de un bloque de árbol en paralelo con la codificación de las filas superiores de bloques de árbol. Este patrón puede continuar por las filas de bloques de árbol en la imagen.

Si el codificador de vídeo está codificando la imagen usando WPP, el codificador de vídeo puede usar información asociada con las CU espacialmente vecinas fuera de un bloque de árbol actual para realizar una intra predicción en una CU particular 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 distinta de la fila más superior, el codificador de vídeo puede usar información asociada con el segundo bloque de árbol de la fila inmediatamente superior para seleccionar un contexto para la codificación CABAC de 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 vídeo puede usar información asociada con un bloque de árbol a la izquierda del bloque de árbol actual para seleccionar un contexto para la codificación CABAC de un elemento de sintaxis del bloque de árbol actual. De esta forma, el codificador de vídeo puede inicializar los estados de CABAC de una fila basándose en los estados CABAC de la fila inmediatamente superior después de codificar dos o más bloques de árbol de la fila inmediatamente superior.

En algunos ejemplos, si el codificador de vídeo está codificando una imagen usando WPP, los únicos límites de pieza de la imagen son los bordes horizontales y verticales de la imagen. Por lo tanto, la única pieza de la imagen puede tener el mismo tamaño que la imagen. El codificador de vídeo puede dividir la imagen y, por tanto, la única pieza de la imagen, en múltiples ondas WPP.

Como se mencionó anteriormente, el codificador 20 de vídeo puede generar una unidad de NAL de corte codificada que incluye una representación codificada de un corte. El corte puede estar asociado con un número entero de bloques de árbol codificados consecutivamente. La unidad de NAL de corte codificada puede incluir un encabezado de corte y datos de corte. Los datos del corte pueden incluir representaciones codificadas de cada bloque de árbol asociado con el corte.

El codificador 20 de vídeo puede generar la unidad de NAL de corte codificada de modo que las representaciones codificadas de los bloques de árbol se agrupan dentro de los datos de corte de acuerdo con las particiones de imagen a las que pertenecen los bloques de árbol. Por ejemplo, la unidad de NAL de corte 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.

Para facilitar la explicación, esta divulgación puede utilizar las expresiones "partición de imagen codificada", "sub-corte codificado", "subflujo codificado" para referirse a un grupo de bloques de árbol codificados dentro de una unidad de NAL de corte codificada que están asociados con una sola partición de imagen (por ejemplo, corte de entropía, pieza u onda WPP). No obstante, debe tenerse en cuenta que puede haber circunstancias en las que una partición de imagen pueda incluir bloques de árbol en múltiples cortes. La Figura 10, que se describe a continuación, ilustra una instancia en la que una pieza incluye bloques de árbol que pertenecen a múltiples cortes y un corte incluye bloques de árbol que pertenecen a múltiples piezas.

El decodificador 30 de vídeo puede recibir un flujo de bits generado por el codificador 20 de vídeo. El flujo de bits puede incluir una representación codificada de datos de vídeo codificados por el codificador 20 de vídeo. El decodificador 30 de vídeo puede realizar una operación de análisis en el flujo de bits. Como parte de la realización de la operación de análisis, el decodificador 30 de vídeo puede extraer elementos de sintaxis del flujo de bits. Como parte de la extracción de elementos de sintaxis del flujo de bits, el decodificador 30 de vídeo puede realizar operaciones de decodificación por entropía (por ejemplo, decodificación de CABAC) sobre datos en el flujo de bits. El decodificador 30 de vídeo puede reconstruir las imágenes de los datos de vídeo basándose en los elementos de sintaxis extraídos del flujo de bits. El proceso para reconstruir los datos de vídeo basándose en los elementos de sintaxis puede ser generalmente recíproco al proceso realizado por el codificador 20 de vídeo para generar los elementos de sintaxis.

El decodificador 30 de vídeo puede generar, basándose en elementos de sintaxis asociados con una CU, bloques de píxeles predictivos para las PU de la CU. Además, el decodificador 30 de vídeo puede cuantificar de forma inversa los bloques de coeficientes asociados con las TU de la CU. El decodificador 30 de vídeo puede realizar transformadas inversas en los bloques de coeficientes para reconstruir bloques de píxeles residuales asociados con las TU de la CU. El decodificador 30 de vídeo puede reconstruir el bloque de píxeles de una CU basándose en los bloques de píxeles predictivos y los bloques de píxeles residuales. De esta forma, el decodificador 30 de vídeo puede reconstruir los bloques de píxeles de las CU basándose en los elementos de sintaxis del flujo de bits.

Como se mencionó anteriormente, algunas implementaciones del decodificador 30 de vídeo pueden decodificar los bloques de píxeles de diferentes particiones de imagen en paralelo. Por ejemplo, si una unidad de NAL de corte codificada incluye una primera partición de imagen codificada y una segunda partición de imagen codificada, el decodificador 30 de vídeo puede decodificar los bloques de píxeles de la primera partición de imagen codificada y los bloques de píxeles de la segunda partición de imagen codificada en paralelo.

El decodificador 30 de vídeo puede almacenar una unidad de NAL de corte codificada en la memoria. La memoria puede ser direccionable por bytes. Debido a que el decodificador 30 de vídeo almacena la unidad de NAL de corte codificada en una memoria direccionable por bytes, el decodificador 30 de vídeo puede ser incapaz de indicar la dirección de memoria asociada con el inicio de una partición de imagen codificada si el inicio de la partición de imagen codificada ocurre dentro de un byte. Por tanto, el decodificador 30 de vídeo puede no ser capaz de decodificar las particiones de imagen codificada en paralelo si una o más de las particiones de imagen codificada comienzan dentro de un byte. Como alternativa, el decodificador 30 de vídeo puede usar direccionamiento de memoria a nivel de bit o direccionamiento a nivel de byte más bit para posibilitar la decodificación de las particiones de imagen codificada en paralelo, pero con mayor complejidad de implementación y cálculo.

Para tratar este problema, el codificador 20 de vídeo puede generar unidades de NAL de corte codificadas de tal manera que cada una de las particiones de imagen codificada dentro de las unidades de NAL de corte codificadas esté alineada en bytes. Es decir, cada una de las particiones de imagen codificada dentro de una unidad de NAL de corte codificada puede comenzar en un límite de bytes. Si una partición de imagen codificada está alineada por bytes, el primer bit de una partición de imagen codificada es el primer bit de un byte direccionable en la memoria y el último bit de la partición de imagen codificada es el último bit de un byte direccionable en la memoria. Por tanto, para cada ubicación de almacenamiento con dirección de byte en la memoria, la ubicación de almacenamiento no almacena bits asociados con múltiples particiones de imagen codificada.

Además, si el decodificador 30 de vídeo decodifica particiones de imagen codificada en paralelo, el decodificador 30 de vídeo puede asignar diferentes particiones de imagen codificada a diferentes núcleos de procesamiento. Los núcleos de procesamiento pueden decodificar las particiones de imagen codificada en paralelo. Además, los núcleos de procesamiento pueden solicitar el encabezado de corte de la memoria. En respuesta, la memoria puede enviar el encabezado de corte a los núcleos de procesamiento. De esta manera, cada uno de los núcleos de procesamiento puede utilizar el encabezado de corte mientras decodifica las particiones de imagen codificada. Sin embargo, si el encabezado de corte no está alineado en bytes, es posible que la memoria no pueda enviar el encabezado de corte a los núcleos de procesamiento sin enviar también información que no está en el encabezado de corte. Por ejemplo, si el encabezado de corte no está alineado en bytes, el decodificador 30 de vídeo puede transcodificar el encabezado de corte de manera que el encabezado de corte esté alineado en bytes. La transcodificación de encabezados de corte puede aumentar la complejidad del decodificador 30 de vídeo y puede requerir que los núcleos de procesamiento se configuren para procesar encabezados de corte transcodificados.

Para generar una unidad de NAL de corte codificada en la que cada una de las particiones de imagen codificada está alineada en bytes, el codificador 20 de vídeo puede rellenar el encabezado de corte de la unidad de NAL de corte codificada de modo que el encabezado de corte finalice en un límite de bytes. Es decir, un último bit de un encabezado de corte puede ser un último bit de un byte direccionable en la memoria. El relleno del encabezado de corte de una unidad de NAL de corte codificada puede implicar la adición de bits al final del encabezado de corte hasta que el encabezado de corte esté alineado en bytes.

De esta manera, el codificador 20 de vídeo puede generar una pluralidad de particiones de imagen codificada, estando asociada cada una de las particiones de imagen codificada con un conjunto diferente de unidades de codificación en una imagen en los datos de vídeo. Además, el codificador 20 de vídeo puede generar una unidad de NAL que contiene un encabezado de corte y la pluralidad de particiones de imagen codificada, incluyendo el encabezado de corte uno o más bits de relleno que alinean el encabezado de corte con un límite de bytes. De manera similar, el decodificador 30 de vídeo puede recibir una unidad de NAL. La unidad de NAL puede contener un encabezado de corte y una pluralidad de particiones de imagen codificada de los datos de vídeo. El encabezado de corte puede incluir uno o más bits de relleno que alinean el encabezado de corte con un límite de bytes. El decodificador 30 de vídeo puede decodificar dos o más de las particiones de imagen codificada en paralelo.

Para decodificar las particiones de imagen codificada en paralelo, el decodificador 30 de vídeo puede asignar las particiones de imagen codificada a diferentes hilos de decodificación que se ejecutan en paralelo. Los subprocesos

de decodificación pueden ejecutarse en diferentes núcleos de procesamiento. Para asignar las particiones de imagen codificada a diferentes hilos de decodificación, el decodificador 30 de vídeo puede necesitar determinar las direcciones de memoria asociadas con los comienzos de las particiones de imagen codificada.

Para permitir que el decodificador 30 de vídeo determine las direcciones de memoria asociadas con los comienzos de las particiones de imagen codificada, el encabezado de corte de la unidad de NAL de corte codificada puede incluir desplazamientos de puntos de entrada. En algunos ejemplos, los desplazamientos del punto de entrada pueden indicar desplazamientos de las direcciones de memoria de las particiones de imagen codificada con respecto a otro punto. En algunos ejemplos, los desplazamientos pueden ser relativos al comienzo de la unidad de NAL de corte codificada. Por ejemplo, un desplazamiento de punto de entrada en un encabezado de corte de una unidad de NAL de corte codificada puede indicar que una partición de imagen codificada particular comienza en un byte de orden 312 de la unidad de NAL de corte codificada. En otro ejemplo, un desplazamiento de punto de entrada para una partición de imagen codificada en una unidad de NAL de corte codificada puede ser relativo a un primer bit de una partición de imagen codificada precedente en la unidad de NAL de corte codificada. Por ejemplo, un desplazamiento de punto de entrada puede indicar que una partición de imagen codificada particular en una unidad de NAL de corte codificada puede comenzar 124 bytes después de una partición de imagen codificada anterior en la unidad de NAL de corte codificada. Además, en algunos ejemplos, los desplazamientos del punto de entrada pueden indicar las longitudes en bits o bytes de las particiones de imagen codificada.

Además de los elementos de sintaxis de desplazamiento que indican los desplazamientos del punto de entrada, el codificador 20 de vídeo puede, de acuerdo con una o más técnicas de esta divulgación, incluir dos elementos de sintaxis adicionales en el encabezado de corte. El número de piezas en el corte se puede determinar basándose en el primero de los elementos de sintaxis adicionales. El número de bits utilizados en cada uno de los elementos de sintaxis de desplazamiento se puede determinar basándose en un segundo de los elementos de sintaxis adicionales. De acuerdo con las técnicas de esta divulgación, el codificador 20 de vídeo puede codificar por entropía (por ejemplo, codificación Exponencial-Golomb) los elementos de sintaxis adicionales. Debido a que la entropía del codificador 20 de vídeo codifica los elementos de sintaxis adicionales, los elementos de sintaxis adicionales se representan en el flujo de bits utilizando secuencias de bits de longitud variable. El uso de secuencias de bits de longitud variable para representar los elementos de sintaxis adicionales puede dar como resultado un flujo de bits que incluye menos bits que cuando se usan secuencias de bits de longitud fija para representar los elementos de sintaxis adicionales, y puede permitir que el codificador 20 de vídeo señalice más elementos de sintaxis de desplazamiento permitidos por la codificación de longitud fija que utiliza, por ejemplo, 5 bits.

De esta manera, el codificador 20 de vídeo puede codificar por entropía un primer elemento de sintaxis, un segundo elemento de sintaxis y una serie de elementos de sintaxis de desplazamiento. Las longitudes de cada uno de los elementos de sintaxis de desplazamiento pueden determinarse basándose en el primer elemento de sintaxis. El número de elementos de sintaxis de desplazamiento en la serie de elementos de sintaxis de desplazamiento puede determinarse basándose en el segundo elemento de sintaxis. Las ubicaciones de una pluralidad de particiones de imagen codificada dentro de una unidad de NAL pueden determinarse basándose en los elementos de sintaxis de desplazamiento. Además, el codificador 20 de vídeo puede generar un flujo de bits que incluye un valor de longitud variable que representa el primer elemento de sintaxis codificado por entropía, un valor de longitud variable que representa el segundo elemento de sintaxis codificado por entropía y valores de longitud fija que representan los elementos de sintaxis de desplazamiento.

El decodificador 30 de vídeo puede recibir un flujo de bits que incluye un valor de longitud variable que representa el primer elemento de sintaxis adicional, un valor de longitud variable codificado por entropía que representa un segundo elemento de sintaxis adicional y una serie de elementos de sintaxis de desplazamiento. El decodificador 30 de vídeo puede decodificar por entropía el primer y segundo elementos de sintaxis adicionales. Además, el decodificador 30 de vídeo puede determinar, basándose en el primer elemento de sintaxis, las longitudes de cada uno de los elementos de sintaxis de desplazamiento y puede determinar, basándose en el segundo elemento de sintaxis, el número de elementos de sintaxis de desplazamiento en la serie de elementos de sintaxis de desplazamiento. El decodificador 30 de vídeo puede analizar, basándose al menos en parte en las longitudes de cada uno de los elementos de sintaxis de desplazamiento y el número de elementos de sintaxis de desplazamiento en la serie de elementos de sintaxis de desplazamiento, los elementos de sintaxis de desplazamiento del flujo de bits. El decodificador 30 de vídeo puede entonces determinar, basándose al menos en parte en los elementos de sintaxis de desplazamiento, las ubicaciones de una pluralidad de particiones de imagen codificada dentro de una unidad de NAL.

Además, en lugar de (o además de) indicar los desplazamientos del punto de entrada en el encabezado de corte de una unidad de NAL de corte codificada, los datos de corte de la unidad de NAL de corte codificada pueden incluir marcadores de punto de entrada. Los marcadores de punto de entrada pueden especificar un patrón de bits que es único de los marcadores de punto de entrada. Los marcadores de punto de entrada pueden aparecer en los datos del corte inmediatamente antes de cada partición de imagen codificada, además de una primera partición de imagen codificada de los datos de corte. El codificador de vídeo 30 puede determinar las direcciones de memoria asociadas con los comienzos de las particiones de imagen codificada escaneando a través de los datos de corte e identificando los marcadores de punto de entrada.

La inclusión de desplazamientos de punto de entrada en el encabezado de corte o marcadores de punto de entrada en los datos de corte puede ser útil si el decodificador 30 de vídeo está configurado para decodificar múltiples particiones de imagen codificada en paralelo. Sin embargo, algunos decodificadores de vídeo no están configurados para decodificar múltiples particiones de imagen codificada en paralelo. El hardware requerido por los decodificadores de vídeo que no están configurados para decodificar múltiples particiones de imagen codificada en paralelo puede ser más sencillo que el hardware requerido por los decodificadores de vídeo que están configurados para decodificar múltiples particiones de imagen codificada en paralelo. El hardware más sencillo puede ser menos costoso y consumir menos energía.

Si el decodificador 30 de vídeo no está configurado para decodificar múltiples particiones de imagen codificada en paralelo, el decodificador 30 de vídeo no usa los desplazamientos de punto de entrada o los marcadores de punto de entrada. Por lo tanto, si el decodificador 30 de vídeo no está configurado para decodificar múltiples particiones de imagen codificada en paralelo, la inclusión de los desplazamientos de punto de entrada y/o los marcadores de punto de entrada en un flujo de bits recibido por el decodificador 30 de vídeo puede aumentar innecesariamente el número de bits en el flujo de bits.

De acuerdo con una o más técnicas de esta divulgación, el codificador 20 de vídeo puede configurarse de manera que el codificador 20 de vídeo no incluya desplazamientos de punto de entrada y/o marcadores de punto de entrada en unidades de NAL de corte codificadas. Más bien, el codificador 20 de vídeo puede generar un mensaje de SEI que indica los puntos de entrada de las particiones de imagen codificada en una unidad de NAL de corte codificada. Por ejemplo, el codificador 20 de vídeo puede generar un mensaje de SEI que incluye desplazamientos de puntos de entrada para las particiones de imagen codificada en una unidad de NAL de corte codificada. En otras palabras, el mensaje de SEI puede indicar los puntos de entrada especificando desplazamientos de bytes de las particiones de imagen codificada dentro de la unidad de NAL de corte codificada.

La Figura 2 es un diagrama de bloques que ilustra un codificador 20 de vídeo de ejemplo que está configurado para implementar las técnicas de esta divulgación. La Figura 2 se proporciona para fines de explicación y no se debería considerar limitante de las técnicas según se ejemplifican y se describen de forma general en esta divulgación. Para fines de explicación, esta divulgación describe el codificador 20 de vídeo en el contexto de la codificación de HEVC. Sin embargo, las técnicas de esta divulgación pueden ser aplicables a otros métodos o normas de codificación.

En el ejemplo de la Figura 2, el codificador 20 de vídeo incluye una pluralidad de componentes funcionales. Los componentes funcionales del codificador 20 de vídeo incluyen una unidad 100 de procesamiento de predicción, una unidad 102 de generación residual, una unidad 104 de procesamiento de transformada, una unidad 106 de cuantificación, una unidad 108 de cuantificación inversa, una unidad 110 de procesamiento de transformada inversa, una unidad 112 de reconstrucción, una unidad 113 de filtro, una memoria intermedia 114 de imágenes decodificadas y una unidad 116 de codificación por entropía. La unidad 100 de procesamiento de predicción incluye una unidad 121 de procesamiento de inter-predicción y una unidad 126 de procesamiento de intra-predicción. La unidad 121 de procesamiento de inter-predicción incluye una unidad 122 de estimación de movimiento y una unidad 124 de compensación de movimiento. En otros ejemplos, el codificador 20 de vídeo puede incluir más, menos o diferentes componentes funcionales. Además, la unidad 122 de estimación de movimiento y la unidad 124 de compensación de movimiento pueden estar altamente integradas, pero están representadas en el ejemplo de la Figura 2 por separado para propósitos de explicación.

El codificador 20 de vídeo puede recibir datos de vídeo. El codificador 20 de vídeo puede recibir los datos de vídeo de diversas fuentes. Por ejemplo, el codificador 20 de vídeo puede recibir los datos de vídeo del origen 18 de vídeo (Figura 1) u otra fuente. Los datos de vídeo pueden representar una serie de imágenes. Para codificar los datos de vídeo, el codificador 20 de vídeo puede codificar cada corte de cada una de las imágenes. Como parte de la codificación de un corte, el codificador 20 de vídeo puede codificar cada bloque de árbol del corte.

El codificador 20 de vídeo puede codificar los bloques de árbol en el corte de acuerdo con un orden de exploración por tramas. En otras palabras, el codificador 20 de vídeo puede codificar los bloques de árbol del corte en un orden que avanza de izquierda a derecha a través de una fila superior de bloques de árbol en el corte, a continuación, continúa de izquierda a derecha a través de la siguiente fila inferior de bloques de árbol, y así sucesivamente hasta que el codificador 20 de vídeo haya codificado cada uno de los bloques de árbol en el corte.

Como resultado de la codificación de los bloques del árbol de acuerdo con el orden de escaneo por tramas, los bloques del árbol arriba y a la izquierda de un bloque de árbol particular pueden haberse codificado, pero los bloques de árbol debajo y a la derecha del bloque de árbol particular aún no se han codificado. En consecuencia, el codificador 20 de vídeo puede acceder a la información generada codificando bloques de árbol arriba y a la izquierda del bloque de árbol particular cuando se codifica el bloque de árbol particular. Sin embargo, es posible que el codificador 20 de vídeo no pueda acceder a la información generada codificando bloques de árbol debajo y a la derecha del bloque de árbol particular cuando se codifica el bloque de árbol particular.

Como parte de la codificación de un bloque de árbol, la unidad 100 de procesamiento de predicción puede realizar subdivisión de árbol cuádruple en el bloque de píxeles asociado con el bloque de árbol para dividir el bloque de píxeles

en bloques de píxeles progresivamente más pequeños. Los bloques de píxeles más pequeños pueden asociarse con las CU. Por ejemplo, la unidad 100 de procesamiento de predicción puede dividir cada uno de los bloques de píxeles de un bloque de árbol en cuatro sub-bloques de igual tamaño, dividir uno o más de los sub-bloques en cuatro sub-bloques de igual tamaño y así sucesivamente.

Los tamaños de los bloques de píxeles asociados con las CU pueden variar desde 8x8 píxeles hasta el tamaño de los bloques de píxeles asociados con los bloques de árbol con un máximo de 64x64 muestras o más. En esta divulgación, "NxN" y "N por N" pueden usarse de manera intercambiable para referirse a las dimensiones de píxeles de un bloque de píxeles en términos de dimensiones verticales y horizontales, por ejemplo, 16x16 píxeles o 16 por 16 píxeles. En general, un bloque de 16x16 píxeles tiene dieciséis píxeles en dirección vertical (y = 16) y dieciséis píxeles en dirección horizontal (x = 16). Análogamente, un bloque NxN tiene en general N píxeles en una dirección vertical y N píxeles en una dirección horizontal, donde N representa un valor entero no negativo.

El codificador 20 de vídeo puede codificar las CU de un bloque de árbol para generar representaciones codificadas de las CU (es decir, CU codificadas). El codificador 20 de vídeo puede codificar las CU de un bloque de árbol según un orden de exploración en z. En otras palabras, el codificador 20 de vídeo puede codificar una CU superior izquierda, una CU superior derecha, una CU inferior izquierda y a continuación una CU inferior derecha, en ese orden. Cuando el codificador 20 de vídeo codifica una CU particionada, el codificador 20 de vídeo puede codificar las CU asociadas con subbloques de los bloques de píxeles de la CU particionada según el orden de exploración en z. En otras palabras, el codificador 20 de vídeo 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 a continuación una CU asociada con un subbloque inferior derecho, en ese orden.

Como resultado de la codificación de las CU de un bloque de árbol de acuerdo con un orden de exploración en z, las CU arriba, arriba y a la izquierda, arriba y a la derecha, izquierda y abajo y a la izquierda de una CU particular pueden haberse codificado. Las CU que se encuentran debajo o a la derecha de la CU en particular aún no se han codificado. En consecuencia, el codificador 20 de vídeo puede acceder a la información generada codificando algunas CU vecinas a la CU particular cuando se codifica la CU particular. Sin embargo, es posible que el codificador 20 de vídeo no pueda acceder a la información generada codificando otras CU vecinas a la CU particular cuando se codifica la CU particular.

Como parte de la codificación de una CU, la unidad 100 de procesamiento de predicción puede dividir los bloques de píxeles de la CU entre una o más PU de la CU. El codificador 20 de vídeo y el decodificador 30 de vídeo pueden soportar diversos tamaños de PU. Suponiendo que el tamaño de una CU particular es de 2Nx2N, el codificador 20 de vídeo y el decodificador 30 de vídeo pueden soportar tamaños de PU de 2Nx2N o NxN para la intra predicción, y tamaños de PU simétricos de 2Nx2N, 2NxN, Nx2N, NxN o similares para la inter predicción. El codificador 20 de vídeo y el decodificador 30 de vídeo también pueden soportar una subdivisión asimétrica para tamaños de PU de 2NxnU, 2NxnD, nLx2N y nRx2N para la inter predicción.

La unidad 121 de procesamiento de inter-predicción puede realizar una inter predicción en cada PU de la CU. La inter predicción puede proporcionar compresión temporal. La unidad 121 de procesamiento de inter-predicción puede generar datos predictivos para una PU. Los datos predictivos para la PU pueden incluir bloques de muestra predictivos que corresponden a la PU y la información de movimiento para la PU. La unidad 122 de estimación de movimiento puede generar la información de movimiento para la PU. En algunos casos, la unidad 122 de estimación de movimiento puede usar el modo de unión o el modo de predicción de vector de movimiento avanzado (AMVP) para señalizar la información de movimiento de la PU. La unidad 124 de compensación de movimiento puede generar los bloques de píxeles predictivos de la PU basándose en los píxeles de una o más imágenes distintas de la imagen asociada con la PU (es decir, imágenes de referencia).

Los cortes pueden ser cortes I, cortes P o cortes B. La unidad 122 de estimación de movimiento y la unidad 124 de compensación de movimiento pueden realizar diferentes operaciones para una PU de una CU dependiendo de si la PU está en un corte I, un corte P o un corte B. En un corte I, todas las PU son intra predichas. Por lo tanto, si la PU está en un corte I, la unidad 122 de estimación de movimiento y la unidad 124 de compensación de movimiento no realizan inter predicción en la PU.

Si la PU está en un corte P, la imagen que contiene la PU se asocia con una lista de imágenes de referencia denominada como "lista 0". En algunos ejemplos, cada imagen de referencia enumerada en la lista 0 aparece antes de la imagen en el orden de visualización. Cada una de las imágenes de referencia en la lista 0 puede usarse para la predicción de otras imágenes. La unidad 122 de estimación de movimiento puede buscar las imágenes de referencia en la lista 0 para un bloque de referencia para una PU en un corte P. El bloque de referencia de la PU puede ser un bloque de píxeles que se corresponda más estrechamente con el bloque de píxeles de la PU. La unidad 122 de estimación de movimiento puede usar una diversidad de métricas para determinar qué tan cerca se corresponde un bloque de píxeles en una imagen de referencia con el bloque de píxeles de una PU. Por ejemplo, la unidad 122 de estimación de movimiento puede determinar qué tan cerca un bloque de píxeles en una imagen de referencia corresponde al bloque de píxeles de una PU por suma de diferencia absoluta (SAD), suma de diferencia cuadrada (SSD) u otras métricas de diferencia.

La unidad 122 de estimación de movimiento puede generar un índice de imagen de referencia que indica la imagen de referencia en la lista 0 que contiene un bloque de referencia de una PU en un corte P y un vector de movimiento que indica un desplazamiento espacial entre la PU y el bloque de referencia. La unidad 122 de estimación de movimiento puede generar vectores de movimiento con diversos grados de precisión. Por ejemplo, la unidad 122 de estimación de movimiento puede generar vectores de movimiento con precisión de un cuarto de píxel, precisión de un octavo de píxel u otra precisión de píxel fraccional. En el caso de precisión fraccional de píxeles, los valores del bloque de referencia pueden interpolarse a partir de valores de píxeles de posición de número entero en la imagen de referencia. La unidad 122 de estimación de movimiento puede generar el índice de imagen de referencia y el vector de movimiento como la información de movimiento de la PU. La unidad 124 de compensación de movimiento puede generar los bloques de muestra predictivos de la PU basándose en el bloque de referencia asociado con la información de movimiento de la PU.

Si la PU está en un corte B, la imagen que contiene la PU puede asociarse con dos listas de imágenes de referencia, denominadas como "lista 0" y "lista 1". En algunos ejemplos, cada imagen de referencia enumerada en la lista 1 aparece después de la imagen actual en el orden de visualización. Además, si la PU está en un corte B, la unidad 122 de estimación de movimiento puede realizar una inter predicción unidireccional o una inter predicción bidireccional para la PU. Para realizar la inter predicción unidireccional para la PU, la unidad 122 de estimación de movimiento puede buscar las imágenes de referencia de la lista 0 o la lista 1 para un bloque de referencia para la PU. La unidad 122 de estimación de movimiento puede generar un índice de imagen de referencia que indica una posición en la lista 0 o lista 1 de la imagen de referencia que contiene el bloque de referencia y un vector de movimiento que indica un desplazamiento espacial entre la PU y el bloque de referencia.

Para realizar la inter predicción bidireccional para una PU, la unidad 122 de estimación de movimiento puede buscar las imágenes de referencia en la lista 0 para un bloque de referencia para la PU y también puede buscar las imágenes de referencia en la lista 1 para otro bloque de referencia para la PU. La unidad 122 de estimación de movimiento puede generar índices de imágenes de referencia que indican posiciones en la lista 0 y la lista 1 de las imágenes de referencia que contienen los bloques de referencia. Además, la unidad 122 de estimación de movimiento puede generar vectores de movimiento que indican desplazamientos espaciales entre los bloques de referencia y la PU. La información de movimiento de la PU puede incluir los índices de imagen de referencia y los vectores de movimiento de la PU. La unidad 124 de compensación de movimiento puede generar los bloques de muestra predictivos de la PU basándose en los bloques de referencia indicados por la información de movimiento de la PU.

Como parte de la realización de una operación de codificación en una CU, la unidad 126 de procesamiento de intra predicción puede realizar intra predicción en las PU de la CU. La intra predicción puede proporcionar compresión espacial. La unidad 126 de procesamiento de intra-predicción puede generar datos predictivos para una PU basándose en muestras decodificadas en la misma imagen que la PU. Los datos predictivos para la PU pueden incluir bloques de muestra predictiva para la PU y diversos elementos de sintaxis. La unidad 126 de procesamiento de intra-predicción puede realizar una intra predicción sobre unas PU en cortes I, cortes P y cortes B.

Para realizar una intra predicción en una PU, la unidad 126 de procesamiento de intra-predicción puede usar múltiples modos de intra predicción para generar múltiples conjuntos de datos predictivos para la PU. Para utilizar un modo de intra predicción para generar un conjunto de datos predictivos para la PU, la unidad 126 de procesamiento de intra-predicción puede extender muestras de bloques de muestra de PU vecinas a través de los bloques de muestra de la PU en una dirección y/o gradiente asociado con el modo de intra predicción. Las PU vecinas pueden estar arriba, arriba y a la derecha, arriba y a la izquierda, o a la izquierda de la PU, asumiendo un orden de codificación de izquierda a derecha, de arriba a abajo para PU, CU y bloques de árbol. La unidad 126 de procesamiento de intra-predicción puede usar diversos números de modos de intra predicción, por ejemplo, 33 modos de intra predicción direccionales. En algunos ejemplos, el número de modos de predicción intra puede depender del tamaño de la PU.

La unidad 100 de procesamiento de predicción puede seleccionar los datos predictivos para las PU de una CU entre los datos predictivos generados por la unidad 121 de procesamiento de inter-predicción para las PU o los datos predictivos generados por la unidad 126 de procesamiento de intra-predicción para las PU. En algunos ejemplos, la unidad 100 de procesamiento de predicción selecciona los datos predictivos para las PU de la CU basándose en métricas de tasa/distorsión de los conjuntos de datos predictivos.

La unidad 100 de procesamiento de predicción puede realizar subdivisión de árbol cuádruple para subdividir el bloque de píxeles residuales de una CU en subbloques. Cada bloque de píxeles residuales no dividido puede asociarse con una TU diferente de la CU. Los tamaños y posiciones de los bloques de píxeles residuales asociados con las TU de una CU pueden basarse o no en los tamaños y posiciones de los bloques de píxeles de 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 píxeles residuales. Las TU de una CU pueden corresponder a nodos hoja del RQT.

Debido a que los píxeles de los bloques de píxeles residuales de las TU comprenden muestras Y, Cb y Cr, cada una de las TU puede asociarse con un bloque de muestra de Y muestras, un bloque de muestra de muestras Cb y un bloque de muestra de muestras Cr. La unidad 102 de generación residual puede generar bloques de muestra residuales para una CU restando bloques de muestra predictivos de PU de la CU de las muestras correspondientes

de los bloques de muestra de la CU.

La unidad 104 de procesamiento de transformada puede generar bloques de coeficientes para cada TU de una CU aplicando una o más transformadas a los bloques de muestra residuales asociados con la TU. Cada uno de los bloques de coeficientes puede ser una matriz 2D de coeficientes. La unidad 104 de procesamiento de transformada puede aplicar diversas transformadas a un bloque de muestra residual asociado con una TU. Por ejemplo, la unidad 104 de procesamiento de transformada puede aplicar una transformada de coseno discreta (DCT), una transformada direccional o una transformada conceptualmente similar al bloque de muestra residual asociado con una TU.

La unidad 106 de cuantificación puede cuantificar los coeficientes en un bloque de coeficientes asociado con una TU. El proceso de cuantificación puede reducir la profundidad de bits asociada con alguno o todos los coeficientes. Por ejemplo, un coeficiente de *n* bits se puede redondear a un coeficiente de *m* bits durante la cuantificación, donde *n* es mayor que *m.* La unidad 106 de cuantificación puede cuantificar un bloque de coeficientes asociado con una TU de una CU basándose en un valor de parámetro de cuantificación (QP) asociado con la CU. El codificador 20 de vídeo puede ajustar el grado de cuantificación aplicado a los bloques de coeficientes asociados con una CU ajustando el valor de QP asociado con la CU.

El codificador 20 de vídeo puede asociar un valor QP con una CU de diversas formas. Por ejemplo, el codificador 20 de vídeo puede realizar un análisis de tasa-distorsión en un bloque de árbol asociado con la CU. En el análisis de tasa-distorsión, el codificador 20 de vídeo puede generar múltiples representaciones codificadas del bloque de árbol codificando el bloque de árbol varias veces con diferentes valores de QP. El codificador 20 de vídeo puede señalizar que un valor de QP particular está asociado con la CU cuando el valor de QP particular está asociado con la CU en una representación codificada del bloque de árbol que tiene una métrica de distorsión y tasa de bits más baja.

La unidad 108 de cuantificación inversa y la unidad 110 de procesamiento de transformada inversa pueden aplicar cuantificación inversa y transformadas inversas a un bloque de coeficientes, respectivamente, para reconstruir un bloque de muestra residual a partir del bloque de coeficientes. La unidad 112 de reconstrucción puede añadir el bloque de muestra residual reconstruido a las muestras correspondientes de uno o más bloques de muestra predictiva generados por la unidad 100 de procesamiento de predicción para producir un bloque de muestra reconstruido asociado con una TU. Reconstruyendo bloques de muestra para cada TU de una CU de esta manera, el codificador 20 de vídeo puede reconstruir los bloques de muestra de la CU.

La unidad 113 de filtro puede realizar una operación de desbloqueo para reducir los artefactos de bloqueo en bloques de muestra asociados con una CU. La memoria intermedia 114 de imágenes decodificadas puede almacenar los bloques de muestra reconstruidos después de que la unidad 113 de filtro realice la una o más operaciones de desbloqueo en los bloques de muestra reconstruidos. La unidad 122 de estimación de movimiento y la unidad 124 de compensación de movimiento pueden usar una imagen de referencia que contiene los bloques de muestra reconstruidos para realizar la inter predicción en las PU de imágenes posteriores. Además, la unidad 126 de procesamiento de intra-predicción puede usar bloques de muestra reconstruidos en la memoria intermedia 114 de imágenes decodificadas para realizar la intra predicción en otras PU en la misma imagen que la CU.

La unidad 116 de codificación por entropía puede recibir datos desde otros componentes funcionales del codificador 20 de vídeo. Por ejemplo, la unidad 116 de codificación por entropía puede recibir bloques de coeficientes desde la unidad 106 de cuantificación y puede recibir elementos de sintaxis desde la unidad 100 de procesamiento de predicción. La unidad 116 de codificación por entropía puede realizar una o más operaciones de codificación por entropía sobre los datos para generar datos codificados por entropía. Por ejemplo, el codificador 20 de vídeo puede realizar una operación de codificación de longitud variable adaptativa al contexto (CAVLC), una operación CABAC, una operación de codificación de longitud de variable a variable (V2V), una operación de codificación aritmética binaria adaptativa al contexto (SBAC) basada en la sintaxis, una operación de codificación por entropía de particionamiento de intervalo de probabilidad (PIPE), codificación de Golomb exponencial u otro tipo de operación de codificación por entropía en los datos. La unidad 116 de codificación por entropía puede generar un flujo de bits que incluye los datos codificados por entropía.

Como parte de la realización de una operación de codificación por entropía en los datos, la unidad 116 de codificación por entropía puede seleccionar un modelo de contexto. Si la unidad 116 de codificación por entropía está realizando una operación de CABAC, el modelo de contexto puede indicar estimaciones de probabilidades de binarios particulares que tienen valores particulares. En el contexto de CABAC, el término "binario" puede usarse para referirse a una versión convertida a binario de un elemento de sintaxis.

Como se analizó anteriormente, el codificador 20 de vídeo puede generar un flujo de bits que incluye una representación codificada de datos de vídeo. La representación codificada de los datos de vídeo puede incluir una pluralidad de unidades de NAL. Las unidades de NAL pueden incluir unidades de NAL de corte codificadas que incluyen representaciones codificadas de cortes. Debido a que un corte puede incluir bloques de árbol asociados con múltiples particiones de imagen (por ejemplo, cortes de entropía, piezas u ondas WPP), una unidad de NAL de corte codificada puede incluir múltiples particiones de imagen codificada. Como se indicó anteriormente, una partición de imagen codificada puede ser un grupo de bloques de árbol codificados dentro de una unidad de NAL de corte

codificada que están asociados con una única partición de imagen.

Cada una de las unidades de NAL de corte codificadas puede incluir un encabezado de corte y datos de corte. Los datos de corte de una unidad de NAL de corte codificada pueden incluir una representación codificada de cada bloque de árbol en un corte. En algunos casos, los bloques de árbol de un corte pueden estar asociados con diferentes particiones de imágenes. Por consiguiente, los datos de corte de una unidad de NAL de corte codificada única pueden incluir bloques de árbol codificados asociados con múltiples particiones de imagen. Como se indicó anteriormente, las expresiones "partición de imagen codificada" y "sub-corte codificado" pueden usarse para hacer referencia a un grupo de bloques de árbol codificados dentro de una unidad de NAL de corte codificada que están asociados con una única partición de imagen.

El decodificador 30 de vídeo puede configurarse para analizar o decodificar múltiples particiones de imagen codificada en paralelo. Para facilitar el análisis o la decodificación simultánea de múltiples particiones de imagen codificada, cada una de las particiones de imagen codificada puede estar alineada en bytes. Es decir, el primer bit de una partición de imagen codificada es el primer bit de un byte direccionable en la memoria y el último bit de la partición de imagen codificada es el último bit de un byte direccionable en la memoria.

Para garantizar que las particiones de imagen codificada en los datos de corte de una unidad de NAL de corte codificada estén alineadas en bytes, el encabezado de corte de una unidad de NAL de corte codificada puede, de acuerdo con las técnicas de esta descripción, terminar en un límite de byte. En algunos casos, tanto el principio como el final del encabezado de corte pueden estar alineados en bytes. Por ejemplo, el codificador 20 de vídeo puede generar unidades de NAL de corte codificadas con encabezados de corte alineados en bytes para cortes que incluyen bloques de árbol en múltiples cortes por entropía alineados en bytes codificados, piezas u ondas WPP (es decir, particiones de imagen codificada).

En algunos ejemplos, el codificador 20 de vídeo puede incluir una bandera (por ejemplo, slice_header_byte_aligned_flag) en un conjunto de parámetros, como un SPS, un PPS o un APS. Si la bandera es igual a 1, los encabezados de corte de cada unidad de NAL de corte codificada que se refiere al conjunto de parámetros están alineados en bytes. En tales ejemplos, si la bandera es igual a 0, los encabezados de corte de cada unidad de NAL de corte codificada que se refiere al conjunto de parámetros pueden estar o no alineados en bytes. En otros ejemplos, la semántica de la bandera puede invertirse. En otros ejemplos, los encabezados de corte de las unidades de NAL de corte codificadas siempre están alineadas en bytes. Esto puede ser equivalente a que la bandera mencionada anteriormente no esté presente y que el valor de la bandera siempre se infiera que es igual a 1.

De esta manera, el codificador 20 de vídeo puede generar y el decodificador 30 de vídeo puede recibir una unidad de NAL que incluye un conjunto de parámetros. En diferentes ejemplos, el conjunto de parámetros puede ser un SPS, un PPS o un APS. El conjunto de parámetros puede incluir un elemento de sintaxis que indica si los encabezados de corte que hacen referencia al conjunto de parámetros están alineados en bytes.

En los ejemplos en los que el codificador 20 de vídeo genera encabezados de corte alineados en bytes, los encabezados de corte pueden ajustarse a diversas sintaxis. Por ejemplo, el codificador 20 de vídeo puede generar encabezados de corte que se ajustan a la sintaxis de ejemplo indicada en la Tabla 1, a continuación.

TABLA 1: Sintaxis de encabezado de corte

slice_header( ) {	Descriptor
…	
if(slice_header_byte_aligned_flag)	
while(!byte_aligned( ))	
bit_equal_to_one	f(1)
}	

Para facilitar la explicación, los elementos de sintaxis existentes se omiten en la sintaxis de ejemplo de la Tabla 1. En la sintaxis de ejemplo de la Tabla 1, bit_equal a un elemento de sintaxis es un bit que tiene un valor igual a uno. La función byte_aligned( ) puede devolver un valor que indica si el encabezado de corte está alineado en bytes.

Además, de acuerdo con las técnicas de esta divulgación, el flujo de bits puede incluir uno o más mensajes de SEI que indican puntos de entrada de particiones de imagen codificada en unidades de NAL de corte codificadas. Para facilitar la explicación, esta descripción puede hacer referencia a un mensaje de SEI que indica puntos de entrada de particiones de imagen codificada en una unidad de NAL de corte codificada como un "mensaje de SEI de punto de entrada".

En algunos ejemplos, los mensajes de SEI incluyen elementos de sintaxis payloadType que especifican los tipos de

carga útil de los mensajes de SEI. En algunos de estos ejemplos, el elemento de sintaxis payloadType de los mensajes de SEI de punto de entrada puede ser distinto de los elementos de sintaxis payloadType de cualquier otro tipo de mensajes de SEI. Por tanto, un mensaje de SEI de punto de entrada puede incluir un elemento de sintaxis que indica que el mensaje de SEI de punto de entrada pertenece a un tipo de mensajes de SEI que indican puntos de entrada de particiones de imagen codificada en unidades de NAL de corte codificadas.

El flujo de bits puede incluir unidades de NAL de SEI que incluyen mensajes de SEI de punto de entrada. En algunos ejemplos, el mensaje de SEI de punto de entrada en una unidad de NAL de SEI puede aplicarse a la unidad de NAL de corte codificada que sigue inmediatamente a la unidad de NAL de SEI. Así, el decodificador 30 de vídeo puede determinar, basándose al menos en parte en una unidad de NAL de SEI que precede inmediatamente en el flujo de bits a una unidad de NAL que contiene la pluralidad de particiones de imagen codificada, que el mensaje de SEI es aplicable a la unidad de NAL que contiene la pluralidad de particiones de imagen codificada. Para facilitar la explicación, la unidad de NAL de corte codificada a la que se aplica el mensaje de SEI puede denominarse en esta descripción como la unidad de NAL de corte codificada del mensaje de SEI de punto de entrada.

En algunos ejemplos, los mensajes de SEI están asociados con niveles que indican las unidades a las que son aplicables los mensajes de SEI. Los mensajes de SEI de punto de entrada pueden ser aplicables a las unidades de NAL de corte codificadas que siguen a las unidades de NAL de SEI que contienen los mensajes de SEI de punto de entrada. Como alternativa, los mensajes de SEI de punto de entrada pueden ser aplicables a los cortes codificados incluidos en las unidades de NAL de corte codificada que siguen a las unidades de NAL de SEI que contienen los mensajes de SEI de punto de entrada. Por tanto, los mensajes de SEI de punto de entrada pueden ser mensajes de SEI de nivel de unidad de NAL o mensajes de SEI de nivel de corte.

Además, en algunos ejemplos, si una unidad de NAL de corte codificada es la primera unidad de NAL de capa de codificación de vídeo (VCL) de una imagen, la unidad de NAL de SEI que contiene un mensaje de SEI de punto de entrada aplicable a la unidad de NAL de corte codificada puede tener un tipo de unidad de NAL igual a 10. De lo contrario, si una unidad de NAL de corte codificada no es la primera unidad de NAL de VCL de una imagen, la unidad de NAL de SEI que contiene un mensaje de SEI de punto de entrada aplicable a la unidad de NAL de corte codificada puede tener un tipo de unidad de NAL igual a 11. "Unidad de NAL de VCL" es un término colectivo para unidades de NAL de corte codificadas y el subconjunto de unidades de NAL que tienen valores reservados de nal_unit_type que se clasifican como unidades de NAL de VCL. Los tipos de unidades de NAL que no son unidades NAL de VCL pueden denominarse unidades de NAL no de VCL. Los tipos de ejemplo de unidades de NAL no de VCL incluyen unidades NAL de SPS, unidades NAL de PPS, unidades de NAL de SEI, etc. nal_unit_type es un elemento de sintaxis en un encabezado de una unidad de NAL que especifica un tipo de unidad de NAL de la unidad de NAL.

El codificador 20 de vídeo puede generar mensajes de SEI de punto de entrada que tienen diversas sintaxis. Por ejemplo, el codificador 20 de vídeo puede generar mensajes de SEI de punto de entrada que tienen la sintaxis ilustrada en la Tabla 2, a continuación.

TABLA 2: Sintaxis de los mensajes de SEI de punto de entrada

entry_offset(payloadSize) {	Descriptor
offset_len_minus10	ue(v)
num_offsets_minus1	ue(v)
for(i = 0; i <= num_substreams_minus1; i++)	
substream - offset [i]	u(v)
}	

En la sintaxis de ejemplo de la Tabla 2, el elemento de sintaxis offset_len_minus10, más 10, especifica la longitud, en bits, del elemento de sintaxis substream_offset[i]. De esta manera, las longitudes de cada uno de los elementos de sintaxis de desplazamiento son iguales a un valor del elemento de sintaxis offset_len_minus10 más 10. Además, en la sintaxis de ejemplo de la Tabla 2, el elemento de sintaxis num_offsets_minus1, más 1, especifica el número de elementos de sintaxis substream_offset[i] en el mensaje de SEI de punto de entrada.

En la sintaxis de ejemplo de la Tabla 2, el elemento de sintaxis substream_offset[i] especifica el i-ésimo desplazamiento, en bytes. La unidad de NAL de corte codificada del mensaje de SEI de punto de entrada puede constar de num_offsets_minus 1 más 2 particiones de imagen codificada, con valores de índice de partición de imagen codificada 0, 1, 2, ..., num_offsets_minus1+1. La partición de imagen codificada 0 puede constar de los bytes 0 a substream_offset[0] - 1, inclusive, de la unidad de NAL de corte codificada del mensaje de SEI de punto de entrada. La partición de imagen codificada *k,* con *k* en el intervalo de 1 a num_offsets_minus1+1, inclusive, puede constar de bytes substream_offset [*k* - 1] a substream_offset [*k*] - 1, inclusive, de la unidad de NAL de corte codificada del mensaje de SEI de punto de entrada.

De esta manera, el codificador 20 de vídeo puede señalar un elemento de sintaxis a partir del cual se pueden determinar las longitudes en bits de los elementos de sintaxis desplazados. Además, el codificador 20 de vídeo puede señalizar un elemento de sintaxis a partir del cual se puede determinar el número de elementos de sintaxis de desplazamiento en la serie de elementos de sintaxis de desplazamiento. Además, el codificador 20 de vídeo puede señalizar la serie de elementos de sintaxis de desplazamiento, en donde los elementos de sintaxis de desplazamiento en la serie de elementos de sintaxis de desplazamiento indican los desplazamientos de bytes de las particiones de imagen codificada dentro de la unidad de NAL. Asimismo, el decodificador 30 de vídeo puede recibir, en el flujo de bits, un elemento de sintaxis que especifica una longitud en bits de una serie de elementos de sintaxis de desplazamiento. El decodificador 30 de vídeo también puede recibir, en el flujo de bits, un elemento de sintaxis que especifica el número de elementos de sintaxis de desplazamiento en la serie de elementos de sintaxis de desplazamiento. Además, el decodificador 30 de vídeo puede recibir, en el flujo de bits, la serie de elementos de sintaxis de desplazamiento, en donde los elementos de sintaxis de desplazamiento en la serie de elementos de sintaxis de desplazamiento indican los desplazamientos de bytes de las particiones de imagen codificada dentro de la unidad de NAL.

En algunas sintaxis de ejemplo distintas de la sintaxis de ejemplo de la Tabla 2, la longitud de substream_offset[i] es uno de 1, 2, 3 y 4 bytes. La longitud del subflujo _offset[i] puede indicarse mediante un elemento de sintaxis que tiene un valor igual a 0, 1, 2 o 3, correspondiente a longitudes de 1, 2, 3 y 4 bytes, respectivamente. En tales ejemplos, este elemento de sintaxis puede reemplazar el elemento de sintaxis offset_len_minus10 de la Tabla 2. Además, en diversos otros ejemplos de sintaxis para mensajes de SEI de punto de entrada, el elemento de sintaxis offset_len_minus10 de la Tabla 2 se reemplaza por el elemento de sintaxis offset_len_minus8 y una longitud mínima para los elementos de sintaxis substream_offset[i] es de ocho bits.

Como se describió anteriormente, se puede permitir la predicción dentro de la imagen a través de particiones de imagen (por ejemplo, cortes de entropía, piezas codificadas u ondas de frente de onda). Por tanto, la decodificación de una partición de imagen codificada puede depender de la decodificación de otra partición de imagen codificada. En algunos ejemplos, las particiones de imagen codificada se encapsulan en diferentes unidades de NAL y las unidades de NAL se encapsulan en diferentes paquetes, por ejemplo, paquetes de protocolo de transporte en tiempo real (RTP). Debido a que diferentes unidades de NAL pueden contener particiones de imagen codificada dependientes, la pérdida de una de estas unidades de NAL puede impedir que el decodificador 30 de vídeo decodifique las particiones de imagen codificada en las otras unidades de NAL. Por tanto, puede ser ventajoso incluir múltiples unidades de NAL contenidas en un paquete. En algunos CÓDEC convencionales, un empaquetador puede ser incapaz de determinar a partir de la información en un encabezado de unidad de NAL de una unidad de NAL de corte codificada que la unidad de NAL de corte codificada contiene una partición de imagen codificada. Más bien, en tales CÓDEC, la información que indica que una unidad de NAL de corte codificada contiene una partición de imagen codificada sólo puede estar presente en un encabezado de corte de la unidad de NAL de corte codificada. El encabezado de corte de una unidad de NAL de corte codificada está codificado por entropía. Por tanto, determinar que una unidad de NAL de corte codificada contiene una partición de imagen codificada puede requerir que el empaquetador decodifique por entropía el encabezado de corte de la unidad de NAL de corte codificada, lo que añade complejidad al empaquetador.

Para reducir la complejidad del empaquetador, se puede asignar un tipo de unidad de NAL distinto de las unidades de NAL que contienen una única partición de imagen codificada para la que se permite la predicción dentro de la imagen a través de los límites de la partición de imagen. Además, se pueden asignar diferentes tipos de unidades de NAL a unidades de NAL de SEI que contienen información con diferentes ámbitos de aplicación.

El codificador 20 de vídeo puede indicar el tipo de unidad de NAL de diversas formas. Por ejemplo, el codificador 20 de vídeo puede incluir un elemento de sintaxis "nal_unit_type" en un encabezado de unidad de NAL de una unidad de NAL para indicar el tipo de unidad de NAL. En algunos ejemplos, el elemento de sintaxis "nal_unit_type" puede ser de seis bits. En tales ejemplos, el valor especificado por el elemento de sintaxis "nal_unit_type" puede oscilar entre 0 y 63. En esta descripción, se supone que los intervalos son inclusivos a menos que se indique lo contrario. Las unidades de NAL con elementos de sintaxis "nal_unit_type" iguales a 0 o en el intervalo de 48 a 63 pueden no afectar al proceso de decodificación. Más bien, las unidades de NAL con elementos de sintaxis "nal_unit_type" iguales a 0 o en el intervalo de 48 a 63 pueden usarse según lo determinado por una aplicación. No se especifica ningún proceso de decodificación para las unidades de NAL con elementos de sintaxis "nal_unit_type" iguales a 0 o en el intervalo de 48 a 63. Diferentes aplicaciones pueden usar unidades de NAL con elementos de sintaxis "nal_unit_type" iguales a 0 o en el intervalo de 48 a 63 para diferentes propósitos. Por lo tanto, se asignan diferentes tipos de unidades de NAL a las unidades de NAL de SEI que contienen información con diferentes ámbitos de aplicación. En otros ejemplos, el elemento de sintaxis "nal_unit_type" puede ser de cinco bits, como en HEVC WD4. En los ejemplos en los que el elemento de sintaxis "nal_unit_type" es de cinco bits, el valor del elemento de sintaxis "nal_unit_type" puede oscilar entre 0 y 31.

La Tabla 3, que se presenta a continuación, ilustra una relación de ejemplo entre los valores del elemento de sintaxis "tipo de unidad final", el contenido de las unidades de NAL y las estructuras de sintaxis de la carga útil de secuencia de bytes sin procesar (RBSP) de las unidades de NAL y las clases de tipo de unidad de NAL.

TABLA 3

nal_unit_type	Contenido de la unidad de NAL y estructura de sintaxis RBSP	Clase de tipo de unidad de NAL
0	Sin especificar	no de VCL
1	Corte codificado de una imagen no de IDR y no de CRA slice_layer_rbsp( )	VCL
2	Sub-corte codificado de una imagen no de IDR y no de CRA slice_layer_rbsp( )	VCL
3	Corte codificado de una imagen CRA slice_layer_rbsp( )	VCL
4	Sub-corte codificado de una imagen CRA slice_layer_rbsp( )	VCL
5	Corte codificado de una imagen de IDR slice_layer_rbsp( )	VCL
6	Sub-corte codificado de una imagen de IDR slice_layer_rbsp( )	VCL
7	Conjunto de parámetros de secuencia seq_parameter_set_rbsp( )	no de VCL
8	Conjunto de parámetros de imagen pic_parameter_set_rbsp( )	no de VCL
9	Conjunto de parámetros de adaptación aps_rbsp( )	no de VCL
10	Información de mejora complementaria (SEI) sei_rbsp( )	no de VCL
11	Información de mejora complementaria (SEI) de la sub-imagen sei_rbsp( )	no de VCL
12	Delimitador de unidad de acceso access_unit_delimiter_rbsp( )	no de VCL
13	Datos de relleno filler_data_rbsp( )	no de VCL
13-47	Reservado	n/d
48..63	Sin especificar	no de VCL

En el ejemplo de la Tabla 3, las unidades de NAL con elementos de sintaxis "nal_unit_type" en el intervalo de 1 a 6 son unidades de NAL de corte codificadas. Si una unidad de NAL contiene datos asociados con un corte o sub-corte de una imagen en particular y la unidad de NAL tiene un elemento de sintaxis "nal_unit_type" igual a 5 o 6, las unidades de NAL con elementos de sintaxis "nal_unit_type" iguales a 1 a 4 no pueden contener datos asociados con la imagen particular. Si una unidad de NAL contiene datos asociados con un corte o sub-corte de una imagen particular y la unidad de NAL tiene un elemento de sintaxis "nal_unit_type" igual a 5 o 6, la imagen puede ser una imagen de refresco de decodificador instantánea (IDR). Como se indica en la Tabla 3, las unidades de NAL que contienen sub-cortes codificados y unidades de NAL de corte codificado que no contienen sub-cortes codificados pueden incluir RBSP que se ajustan a la misma sintaxis.

Si una unidad de NAL tiene un elemento de sintaxis "nal_unit_type" igual a 2, 4 o 6, la unidad de NAL contiene un sub-corte. Por ejemplo, una unidad de NAL puede tener un elemento de sintaxis "nal_unit_type" con un valor de 2, 4 o 6 si la unidad de NAL contiene datos asociados con un corte que tiene un "light_slice_flag" igual a 1 (es decir, un corte ligero). Para facilitar la explicación, las unidades de NAL que tienen elementos de sintaxis "nal_unit_type" iguales a 2, 4 o 6 pueden denominarse en el presente documento unidades de NAL de sub-corte. Por tanto, las unidades de NAL que contienen sub-cortes codificados pueden pertenecer a un tipo de unidad de NAL diferente al de las unidades de NAL de corte codificado que no contienen sub-cortes codificados. Por ejemplo, las unidades de NAL que contienen los sub-cortes codificados pueden incluir elementos de sintaxis de tipo que especifican un número que es diferente de los números especificados por los elementos de sintaxis de tipo de las unidades de NAL de corte codificadas que no contienen sub-cortes codificados. Durante la decodificación, el decodificador 30 de vídeo puede requerir datos de otro corte o sub-corte para realizar la predicción dentro de la imagen dentro de un sub corte.

Si una unidad de NAL tiene un elemento de sintaxis "nal_unit_type" que es igual a 10 u 11, la unidad de NAL es una unidad de NAL de SEI. Una unidad de NAL de SEI con un elemento de sintaxis "nal_unit_type" igual a 10 puede contener un mensaje de SEI que es aplicable a una o más imágenes codificadas completas. Una unidad de NAL de SEI con elementos de sintaxis "nal_unit_type" iguales a 10 puede preceder a la primera unidad de NAL de VCL de una imagen codificada.

Una unidad de NAL de SEI con un elemento de sintaxis "nal_unit_type" igual a 11 puede denominarse en el presente documento una unidad de NAL de SEI de sub-imagen. Una unidad de NAL de SEI de sub-imagen puede contener un mensaje de SEI que es aplicable a una lista de una o más unidades NAL de VCL consecutivas de una imagen codificada. La lista puede incluir un verdadero subconjunto de todas las unidades NAL de VCL de la imagen codificada y no incluye la primera unidad de NAL de VCL de la imagen codificada. Una unidad de NAL de SEI de sub-imagen puede preceder a la primera unidad de NAL de VCL de la lista de unidades NAL de VCL a las que se aplican los mensajes de SEI incluidos en la unidad de NAL de SEI de sub-imagen.

En otros ejemplos, los valores del elemento de sintaxis "nal_unit_type" pueden tener significados diferentes a los indicados en la Tabla 3. Por ejemplo, en otros ejemplos, los intra cortes (es decir, cortes I) y los cortes intermedios (es decir, cortes P y cortes B) pueden tener diferentes tipos de unidades de NAL. En otras palabras, puede haber un primer valor del elemento de sintaxis "nal_unit_type" para cortes I y un segundo valor del elemento de sintaxis "nal_unit_type" para cortes P y cortes B. En otros ejemplos más, puede haber un primer valor del elemento de sintaxis "nal_unit_type" para los cortes I, un segundo valor del elemento de sintaxis "nal_unit_type" para los cortes P y un tercer valor del elemento de sintaxis "nal_unit_type" para los cortes B. En otros ejemplos más, puede haber diferentes valores del elemento de sintaxis "nal_unit_type" para las unidades de NAL que contienen datos que representan cortes cortos (es decir, cortes con el elemento de sintaxis "light_slice_header" igual a 1) y unidades de NAL que contienen datos que representan cortes completos (es decir, cortes con el elemento de sintaxis "light_slice_header" igual a 0).

En otro ejemplo alternativo a la Tabla 3, los mensajes de SEI pueden clasificarse en niveles. Cada uno de los niveles puede estar asociado con un valor diferente del elemento de sintaxis "nal_unit_type". Los niveles de los mensajes de SEI pueden incluir:

1) Nivel de secuencia: aplicable a una o más secuencias de vídeo codificadas completas.
2) Nivel de sub-secuencia: aplicable a más de una imagen codificada, pero a menos de una secuencia de vídeo codificada.
3) Nivel de imagen: aplicable a todas las unidades NAL de VCL en una unidad de acceso.
4) Nivel de sub-imagen: aplicable a un verdadero subconjunto de todas las unidades NAL de VCL en una unidad de acceso. Por ejemplo, los mensajes de SEI en el nivel de sub-imagen pueden ser aplicables a las unidades NAL de VCL que cubren una cierta región de interés, la unidad de NAL de VCL de una representación de capa como se define en el anexo G de la norma H.264/AVC, las unidades NAL de VCL de un componente de visualización como se define en el anexo H de la norma H.264/AVC, o las unidades NAL de VCL de una imagen codificada redundante como se definen por la norma H.264/AVC.

Un flujo de bits se puede dividir en una serie de unidades de acceso. Cada una de las unidades de acceso puede contener una imagen codificada y cero o más unidades de NAL no VCL. La primera unidad de acceso del flujo de bits empieza con la primera unidad de NAL del flujo de bits. La primera de cualquiera de las siguientes unidades de NAL después de la última unidad de NAL de VCL de una imagen codificada puede especificar el comienzo de una nueva unidad de acceso.

- Unidad de NAL del delimitador de unidad de acceso (cuando está presente).
- Unidad de NAL del conjunto de parámetros de secuencia (cuando está presente).
- Unidad de NAL del conjunto de parámetros de imagen (cuando está presente).
- Unidad de NAL del conjunto de parámetros de adaptación (cuando está presente).
- Unidad de NAL de SEI con "nal_unit_type" igual a 10 (cuando está presente).
- Unidades de NAL con "nal_unit_type" en el intervalo de 14 a 29 (cuando están presentes).
- Primera unidad de NAL de VCL de una imagen codificada (siempre presente).

El codificador 20 de vídeo y el decodificador 30 de vídeo pueden obedecer las siguientes restricciones con respecto al orden de las imágenes codificadas y las unidades de NAL no de VCL dentro de una unidad de acceso:

- Cuando está presente una unidad de NAL de delimitador de la unidad de acceso, el delimitador de la unidad de acceso NAL es la primera unidad de NAL. Solo puede haber una unidad de NAL de delimitador de unidad de acceso en cualquier unidad de acceso.
- Cuando una unidad de NAL de SEI tiene un elemento de sintaxis "nal_unit_type" igual a 10, la unidad de NAL de SEI precede a la primera unidad de NAL de VCL de la imagen codificada.
- Cuando una unidad de NAL de SEI tiene un elemento de sintaxis "nal_unit_type" igual a 11, la unidad de NAL de SEI sucede a la primera unidad de NAL de VCL de la imagen codificada y precede a la primera unidad de NAL de VCL a la que se aplica la unidad de NAL de SEI. Por tanto, la unidad de NAL de SEI precede a la última unidad de

NAL de VCL de la imagen codificada.

- Cuando una unidad de NAL de SEI contiene un mensaje de SEI de período de almacenamiento en memoria intermedia, la unidad de NAL de SEI puede tener un elemento de sintaxis "nal_unit_type" igual a 10 y el mensaje de SEI del período de almacenamiento en memoria intermedia es la carga útil del primer mensaje de SEI de la primera unidad de NAL de SEI en la unidad de acceso.
- Las unidades de NAL que tienen elementos de sintaxis "nal_unit_type" iguales a 0, 11, 13 o en el intervalo de 30 a 63 no preceden a la primera unidad de NAL de VCL de la imagen codificada.

Las unidades de NAL de SPS, las unidades NAL de PPS o las unidades de NAL de APS pueden estar presentes en una unidad de acceso, pero pueden no seguir la última unidad de NAL de VCL de la imagen codificada primaria dentro de la unidad de acceso.

De esta manera, el codificador 20 de vídeo puede implementar un método para codificar datos de vídeo. El método puede comprender generar una pluralidad de unidades de NAL. El método también puede comprender identificar un subconjunto de las unidades de NAL que contienen sub-cortes codificados basadas en tipos de unidades de NAL de las unidades de NAL. Además, el método puede comprender generar un paquete que contiene el subconjunto de las unidades de NAL. De manera similar, el decodificador 30 de vídeo puede implementar un método para decodificar datos de vídeo que comprende recibir un paquete que contiene múltiples unidades de NAL, cada una de las unidades de NAL contiene sub-cortes codificados, cada una de las unidades de NAL incluye un indicador de tipo que indica que las unidades de NAL pertenecen a un tipo de unidades de NAL que contienen sub-cortes codificados (es decir, particiones de imagen codificada). El método para decodificar los datos de vídeo también puede comprender decodificar los sub-cortes.

En otros ejemplos, el codificador 20 de vídeo puede generar una única unidad de NAL que contiene datos asociados con múltiples sub-cortes. Por ejemplo, el codificador 20 de vídeo puede generar una unidad de NAL de un solo corte codificado que contiene datos asociados con múltiples cortes ligeros. En tales ejemplos, los cortes ligeros dentro de la unidad de NAL pueden estar alineados en bytes y el codificador 20 de vídeo puede señalizar desplazamientos de bytes de los datos asociados con cada uno de los sub-cortes, además de los datos asociados con el primero de los sub-cortes. El decodificador 30 de vídeo puede usar los desplazamientos de bytes para asignar diferentes sub-cortes a diferentes núcleos de procesamiento para la decodificación. De esta manera, el decodificador 30 de vídeo puede decodificar los datos asociados con los sub-cortes en paralelo. Sin embargo, si el decodificador 30 de vídeo no decodifica sub-cortes en paralelo, el decodificador 30 de vídeo puede ignorar los desplazamientos en bytes. En consecuencia, si el decodificador 30 de vídeo no decodifica sub-cortes en paralelo, puede ser ineficaz enviar los desplazamientos en bytes.

En algunos ejemplos, las piezas se pueden utilizar para realizar un refresco de decodificación gradual (GDR). Por ejemplo, el codificador 20 de vídeo puede subdividir cada imagen en una secuencia de imágenes en piezas de la misma manera. Por ejemplo, el codificador 20 de vídeo puede utilizar los mismos nueve límites de piezas verticales para subdividir cada una de las imágenes en ocho piezas. En este ejemplo, la pieza más a la izquierda puede ser la pieza 0, la segunda pieza de la izquierda puede ser la pieza 1, y así sucesivamente. Cada una de las imágenes contiene al menos un corte P. En consecuencia, al menos parte de cada imagen de la secuencia depende de al menos otra imagen. En la primera imagen de la secuencia (es decir, la imagen 0), todos los bloques de árbol en la pieza 0 están intra-codificados y los bloques de árbol en todas las piezas restantes de la primera imagen están inter-codificados. En la segunda imagen de la secuencia (es decir, la imagen 1), todos los bloques de árbol en la pieza 1 están intra codificados y los bloques de árbol en todas las piezas restantes de la segunda imagen están inter-codificados. Este patrón puede continuar a través de las imágenes en la secuencia. En otras palabras, en la imagen N de la secuencia de imágenes, todos los bloques de árbol en la pieza N/8 están intra-codificados y los bloques de árbol en todas las demás piezas están inter-codificados, para cualquier valor N en el intervalo de 0 al número de imágenes en la secuencia menos uno. En la oración anterior, "/" indica división modular (es decir, el operador de módulo). Cada imagen con un valor de índice N para el que N/8 es igual a 0 se puede utilizar como un punto de acceso aleatorio en el sentido de que, si la decodificación empieza desde la imagen, todas las imágenes excepto las siguientes siete imágenes se pueden decodificar por completo. En otros ejemplos, se pueden usar otros números de piezas. Por ejemplo, el codificador 20 de vídeo puede implementar GDR con solo dos piezas por imagen.

El acceso aleatorio se refiere a la decodificación de un flujo de bits de vídeo codificado empezando desde cualquier imagen codificada, particularmente de una imagen codificada que no es la primera imagen codificada en el flujo de bits de vídeo. Una imagen codificada picP es un punto de acceso aleatorio (RAP) si todas las imágenes codificadas que siguen a una imagen codificada picQ tanto en el orden de decodificación como en el orden de salida se pueden decodificar correctamente cuando la decodificación comienza desde picP. picQ puede o no ser lo mismo que picP. Un RAP puede ser una imagen de IDR o una imagen no de IDR. Todas las imágenes codificadas que siguen a una imagen de IDR en el orden de decodificación se pueden decodificar sin inter predicción de cualquier imagen que preceda a la imagen de IDR en el orden de decodificación. Una imagen que no es IDR es cualquier imagen que no sea una imagen de IDR.

Un punto de acceso aleatorio no de IDR es una imagen codificada para la que se cumplen las siguientes condiciones:

- El punto de acceso aleatorio no de IDR (picR) no es una imagen de IDR.
- Siendo el recuento del orden de las imágenes (POC) del picR rPoc, siendo picA una imagen que sigue a picR en la misma secuencia de vídeo codificada tanto en el orden de decodificación como en el orden de salida, y siendo POC de picA aPoc. Cuando se realiza un acceso aleatorio en picR (es decir, cuando la decodificación empieza desde picR), todas las imágenes que están en la misma secuencia de vídeo codificada y siguen una imagen picA en el orden de salida se pueden decodificar correctamente.

Un RAP no de IDR es una imagen de acceso aleatorio limpio (CRA) si todas las imágenes que están en la misma secuencia de vídeo codificada que el RAP no de IDR y siguen el RAP no de IDR en el orden de salida se pueden decodificar correctamente cuando se realiza el acceso aleatorio en el RAP no de IDR. Si esto no es cierto, el RAP no de IDR puede denominarse imagen de refresco de decodificación gradual (GDR).

Cada imagen de una secuencia de GDR puede tener su propio PPS. Cada uno de los PPS especifica los límites de las piezas. En otras palabras, los elementos de sintaxis que indican los límites de las piezas se pueden replicar en los PPS para cada imagen de la secuencia GDR. Esto puede ser un desperdicio de bits y puede imponer la necesidad de actualizaciones de PPS en banda, lo que puede imponer problemas de resistencia a errores.

En algunos ejemplos, el codificador 20 de vídeo puede posibilitar un límite de pieza cambiante donde las imágenes solo incluyen dos piezas. Por ejemplo, el codificador 20 de vídeo puede subdividir cada imagen en una secuencia de imágenes en un par de piezas. En este caso, el codificador 20 de vídeo puede permitir un límite de pieza cambiante solo para imágenes con un único límite de pieza vertical u horizontal que no está en un límite de imagen. El límite de la pieza puede moverse de imagen a imagen. Por ejemplo, si el límite de la pieza es vertical, el límite de la pieza puede moverse de izquierda a derecha o de derecha a izquierda. Si el límite de la pieza es horizontal, el límite de la pieza puede moverse hacia arriba o hacia abajo.

El movimiento del límite de la loseta puede señalizarse de diversas formas. Por ejemplo, el codificador 20 de vídeo puede generar uno o más elementos de sintaxis que indican el movimiento del límite de la pieza de imagen a imagen dentro de la secuencia. Dichos elementos de sintaxis pueden estar ubicados dentro de varias estructuras de sintaxis. Por ejemplo, el codificador 20 de vídeo puede generar un conjunto de parámetros que incluye al menos algunos de los elementos de sintaxis que indican el movimiento y la posición del límite de la pieza. En este ejemplo, el conjunto de parámetros puede ser un SPS, un PPS, un APS u otro tipo de conjunto de parámetros. En algunos de estos ejemplos, el movimiento y la posición del límite de la pieza pueden indicarse en combinación con elementos de sintaxis en un SPS y/o un PPS. En otros ejemplos, el movimiento y la posición del límite de la pieza solo se indican mediante uno o más elementos de sintaxis en el encabezado de APS o de corte. En otros ejemplos, el codificador 20 de vídeo puede generar un encabezado de corte que comprende uno o más elementos de sintaxis que indican el movimiento y la posición del límite de la pieza.

En otro ejemplo más, los elementos de sintaxis en un SPS y un encabezado de corte indican el movimiento y la posición del límite de la pieza. En este ejemplo, la Tabla 4 proporciona una sintaxis para un RBSP de SPS. La Tabla 5 proporciona una sintaxis para un encabezado de corte. Juntos, los elementos de sintaxis en el RBSP de SPS de la Tabla 4 y el encabezado de corte de la Tabla 5 señalizan el movimiento del límite de la pieza.

TABLA 4

seq_parameter_set_rbsp( ) {	Descriptor
profile_idc	u(8)
reserved_zero_8bits /* igual a 0 */	u(8)
level_idc	u(8)
seq_parameter_set_id	ue(v)
pic_width_in_luma_samples	u(16)
pic_height_in_luma_samples	u(16)
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
bit_depth_luma_increment	ue(v)
bit_depth_chroma_increment	ue(v)
log2_max_frame_num_minus4	ue(v)

(continuación)

seq_parameter_set_rbsp( ) {	Descriptor
pic_order_cnt_type	ue(v)
if(pic_order_cnt_type = = 0)	
log2_max_pic_order_cnt_lsb_minus4	ue(v)
else if(pic_order_cnt_type = = 1) {	
delta_pic_order_always_zero_flag	u(1)
offset_for_non_ref_pic	se(v)
num_ref_frames_in_pic_order_cnt_cycle	ue(v)
for(i = 0; i<num_ref_frames_in_pic_order_cnt_cycle; i++)	
offset_for_ref_frame[i]	se(v)
}	
max_num_ref_frames	ue(v)
gaps_in_frame_num_value_allowed_flag	u(1)
log2_min_coding_block_size_minus3	ue(v)
log2_diff_max_min_coding_block_size	ue(v)
log2_min_transform_block_size_minus2	ue(v)
log2_diff_max_min_transform_block_size	ue(v)
max_transform_hierarchy_depth_inter	ue(v)
max_transform_hierarchy_depth_intra	ue(v)
interpolation_fllter_flag	u(1)
num_tile_columns_minus1	ue(v)
num_ti le_rows_minus1	ue(v)
if(num_tile_columns_minus1 != 0 || num_tile_rows_minus 1 != 0) {	
tile_boundary_independence_idc	u(1)
if(num_tile_columns_minus1 + num_tile_rows_minus1 = = 1) // uno es igual a 0 y el otro es igual a 1	
dynamic_tile boundary_flag	u(1)
if(dynamic_tile_boundary_flag) {	
tile_boundary_change_direction_flag	u(1)
tile_boundary_change_rate_minus1	ue(v)
}	
else {	
uniform_spacing_idc	u(1)
if(uniform_spacing_idc != 1) {	
for(i=0; i<num_tile_columns_minus1 ; i++)	
column_width[i]	ue(v)
for(i=0; i <num_tile_rows_minus1; i++)	
row_height[i]	ue(v)

(continuación)

seq_parameter_set_rbsp( ) {	Descriptor
}	
}	
}	
rbsp_trailing_bits( )	
}	

TABLA 5

slice_header( ) {	Descriptor
lightweight_slice_flag	u(1)
if(!lightweight_slice_flag) {	
slice_type	ue(v)
pic_parameter_set_id	ue(v)
frame_num	u(v)
if(IdrPicFlag)	
idr_pic_id	ue(v)
if(pic_order_cnt_type = = 0)	
pic_order_cnt_lsb /*	u(v)
if(slice_type = = P || slice_type = = B) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag) {	
num_ref_idx_10_active_minus1	ue(v)
if(slice_type = = B)	
num_ref_idx_l1_active_minus1	ue(v)
}	
}	
ref_pic_list_modification( )	
ref_pic_list_combination( )	
if(nal_ref_idc != 0)	
dec_ref_pic_marking( )	
}	
if(entropy_coding_mode_flag && slice_type != I)	
cabac_init_idc	ue(v)
first_slice_in_pic_flag	u(1)
if(first_slice_in_pic_flag == 0)	
slice_address	u(v)
if(!lightweight_slice_flag) {	

(continuación)

slice_header( ) {	Descriptor
slice_qp_delta	se(v)
if(sample_adaptive_offset_enabled_flag)	
sao_param( )	
if(deblocking_filter_control_present_flag) {	
disable_deblocking_filter_idc	
if(disable_deblocking_filter_idc != 1) {	
slice_alpha_c0_offset_div2	
slice_beta_offset_div2	
}	
}	
if(slice_type == B)	
collocated_from_l0_flag	u(1)
if(adaptive_loop_filter_enabled_flag) {	
if(!shared_pps_info_enabled_flag)	
alf_param( )	
alf_cu_control_param( )	
}	
if(dynamic_tile_boundary_flag)	
tile_boundary_change_cycle	u(v)
}	
}	

En la Tabla 4, el elemento de sintaxis "dynamic_tile_boundary_flag" indica si el límite de la pieza puede cambiar dinámicamente. Por ejemplo, si el elemento de sintaxis "dynamic_tile_boundary_flag" es igual a 1, el límite de la pieza puede cambiar dinámicamente. Si el elemento de sintaxis "bandera de límite de pieza dinámico" es igual a 0, el límite de pieza no cambia dinámicamente. El elemento de sintaxis "bandera de límite de pieza dinámico" puede no ser igual a 1 si la suma de los elementos de sintaxis "num_tile_columns_minus1" y "num_tile_rows_minus1" no es igual a 1. Si el elemento de sintaxis "dynamic_tile_boundary_flag" no está presente, el decodificador 30 de vídeo puede inferir que el elemento de sintaxis "dynamic_tile_boundary_flag" es igual a 0. El elemento de sintaxis "dynamic_tile_boundary_flag" puede ser igual a 1 solo si solo hay dos piezas verticales o dos piezas horizontales.

Además, el elemento de sintaxis "tile_boundary_change_direction flag" de la Tabla 4 puede especificar una dirección de movimiento del límite de la pieza. Por ejemplo, si el elemento de sintaxis "tile_boundary_change _direction_flag" es igual a 0 y el límite de la pieza es vertical, el límite de la pieza puede moverse de izquierda a derecha de una imagen a la siguiente. Si el elemento de sintaxis "tile_boundary_change_direction_flag" es igual a 1 y el límite de la pieza es horizontal, el límite de la pieza puede moverse de derecha a izquierda de una imagen a la siguiente. Si el elemento de sintaxis "tile_boundary_change_direction_flag" es igual a 0 y el límite de la pieza es horizontal, el límite de la pieza puede moverse de arriba a abajo. Si el elemento de sintaxis "tile_boundary_change_direction_flag" es igual a 1 y el límite de la pieza es horizontal, el límite de la pieza puede moverse de abajo hacia arriba.

El elemento de sintaxis "tile_boundary_changeJate_minus1" de la Tabla 4 puede especificar el número de filas o columnas de bloques de árbol que el límite de la pieza mueve de una imagen a la siguiente. Por ejemplo, si el límite de la pieza es vertical, el elemento de sintaxis "tile_boundary_changeJate_minus1" puede indicar el número de columnas de bloques de árbol que el límite de la pieza se mueve de una imagen a la siguiente. Si el límite de la pieza es horizontal, el elemento de sintaxis "tile_boundary_change_rate_minus1" puede indicar el número de filas de bloques de árbol que el límite de la pieza se mueve de una imagen a la siguiente. El número de filas o columnas puede ser igual al elemento de sintaxis "tile_boundary_change_rate_minus1" más 1. Por ejemplo, el valor de la variable "TileBoundaryChangeRate" puede derivarse agregando 1 al elemento de sintaxis "tile_boundary_change_rate_minus1". Para facilitar la explicación, el elemento de sintaxis

"tile_boundary_change_rate_minus1" puede denominarse elemento de sintaxis de tasa.

El decodificador 30 de vídeo puede utilizar el elemento de sintaxis "tile_boundary_change_cycle" de la Tabla 5 para derivar la posición del límite de la pieza. Para facilitar la explicación, el elemento de sintaxis "tile_boundary_change_cycle" puede denominarse como el elemento de sintaxis de ciclo. El elemento de sintaxis de ciclo puede indicar un ciclo asociado con una imagen en la secuencia. Si el límite de la pieza es vertical, el elemento de sintaxis "tile_boundary_change_cycle" puede estar representado por un número de bits indicado por la siguiente fórmula:

Ceil(Log2(PicWidthInLcus + TileBoundaryChangeRate + 1))

En la fórmula anterior, Ceil(...) es la función de techo, Log2(...) es la función logarítmica en base 2 y "PicWidthInLcus" es la anchura de la imagen en LCU (es decir, bloques de árbol). Además, si el límite de la pieza es vertical, el valor del elemento de sintaxis "tile_boundary_change_cycle" puede variar entre 0 y un valor indicado por la siguiente fórmula:

Ceil(PicWidthInLcus + TileBoundaryChangeRate).

Si el límite de la pieza es horizontal, el elemento "tile_boundary_change_cycle" puede estar representado por un número de bits indicado por la siguiente fórmula:

Ceil(Log2(PicHeightInLcus + TileBoundaryChangeRate + 1))

En la fórmula anterior, "PicHeightInLcus" es la altura de la imagen en LCU (es decir, bloques de árbol). Además, si el límite de la pieza es horizontal, el valor del elemento de sintaxis "tile_boundary_change_cycle" puede variar entre 0 y un valor indicado por la siguiente fórmula:

Ceil(PicHeightlnLcus ÷ TileBoundaryChangeRate).

Si el límite de la pieza es vertical (es decir, el elemento de sintaxis "num_tile_columns_minus1" es igual a 1), el decodificador 30 de vídeo puede derivar la posición del límite de la pieza como sigue. Si el elemento de sintaxis "tile_boundary_change_direction_flag" es igual a 0, la anchura en bloques de árbol de la pieza izquierda es igual a "tile_boundary_change_cycle" * "TileBoundaryChangeRate". La variable "TileBoundaryChangeRate" es igual al elemento de sintaxis "tile_boundary_change_rate_minus1" más 1. De lo contrario, si el elemento de sintaxis "tile_boundary_change_direction_flag" es igual a 1, la anchura en bloques de árbol de la pieza derecha es igual a "tile_boundary_change_cycle" * "TileBoundaryChangeRate".

Si el límite de la pieza es horizontal (es decir, el elemento de sintaxis "num_tile_rows_minus1" es igual a 1), el decodificador 30 de vídeo puede derivar la posición del límite de la pieza como sigue. Si el elemento de sintaxis "tile boundary change _direction _flag" es igual a 0, la altura en bloques de árbol de la pieza superior es igual a "tile_boundary_change_cycle" * "TileBoundaryChangeRate". Si el elemento de sintaxis "tile_boundary_change_direction_flag" es igual a 1, la altura en bloques de árbol de la pieza inferior es igual a "tile_boundary_change_cycle" * "TileBoundaryChangeRate".

En la Tabla 5, la variable "IdrPicFlag" se especifica como

IdrPicFlag = ((nal_unit_type = = 5 || nal_unit_type = =6) ? 1: 0).

En otras palabras, la variable "IdrPicFlag" tiene un valor de 1 si el elemento de sintaxis "nal_unit_type" de la unidad de NAL que contiene el encabezado de corte tiene un valor de 5 o 6.

En otro ejemplo, la Tabla 6 ilustra una sintaxis de ejemplo de un conjunto de parámetros de imagen RBSP. Juntos, los elementos de sintaxis del RBSP de PPS y el encabezado de corte de la Tabla 5 pueden señalizar el movimiento y la posición del límite de la pieza.

TABLA 6

pic_parameter_set_rbsp( ) {	Descriptor
pic_paramete_set_id	ue(v)
seq_parameter_set_id	ue(v)
entropy_coding_mode_flag	u(1)
num_ref_idx_l0_default_active_minus1	ue(v)

(continuación)

pic_parameter_set_rbsp( ) {	Descriptor
num_ref_idx_l1_default_active_minus1	ue(v)
picinit_qp_minus26 /*relativo a 26*/	se(v)
constrained_intra_pred_flag	u(1)
tile_info_present_flag	u(1)
if(tile_info-present_flag = = 1) {	
num_tile_columns_minus1	ue(v)
num_ti le_rows_minus1	ue(v)
if(num_tile_columns_minus1 != 0 || num_tile_rows_minus1 != 0) {	
tile_boundary_independence_idc	u(1)
if(num_tile_columns_minus1 + num_tile_rows_minus1 = = 1) // uno es igual a 0 y el otro es igual a 1	
dynamic_tile_boundary_flag	u(1)
if(dynamic_tile_boundary_flag) {	
tile_boundary_change_direction flag	u(1)
tile_boundary_change_rate_minus1	ue(v)
}	
else {	
uniform_spacing_idc	u(1)
if(uniform_spacing_idc != 1) {	
for(i=0; i<num_tile_columns_minus1 ; i++)	
column_width[i]	ue(v)
for(i=0; i <num_tile_rows_minus1; i++)	
row_height[i]	ue(v)
}	
}	
}	
}	
rbsp_trailing_bits( )	
}	

El elemento de sintaxis "dynamic_tile_boundary_flag", el elemento de sintaxis "tile_boundary_change_direction_flag" y el elemento de sintaxis "tile_boundary_change_rate_minus1" de la Tabla 6 pueden tener la misma semántica que los elementos correspondientes de la Tabla 4. Los otros elementos de sintaxis de las Tablas 4, 5 y 6 pueden tener la semántica definida en el documento JCTVC JCTVC-F335 mencionado anteriormente o en HEVC WD4. En algunos ejemplos, el codificador 20 de vídeo y el decodificador 30 de vídeo pueden utilizar la sintaxis de la Tabla 6 además de o como alternativa a la sintaxis de la Tabla 4.

De esta forma, el codificador 20 de vídeo puede realizar un método para codificar datos de vídeo. El método para codificar datos de vídeo puede comprender subdividir cada imagen en una secuencia de imágenes en un par de piezas, separando un límite de pieza el par de piezas. El método de codificación también puede comprender la generación de uno o más elementos de sintaxis que indican el movimiento del límite de la pieza de una imagen a otra dentro de la secuencia. Asimismo, el decodificador 30 de vídeo puede realizar un método para decodificar datos de vídeo. El método de decodificación puede comprender recibir uno o más elementos de sintaxis en un flujo de bits de

los datos de vídeo, indicando el uno o más elementos de sintaxis el movimiento de un límite de pieza de una imagen a otra dentro de una secuencia de imágenes. El método de decodificación también puede comprender determinar, para cada una de las imágenes, una posición del límite de la pieza basándose en los elementos de sintaxis.

La Figura 3 es un diagrama de bloques que ilustra un decodificador 30 de vídeo de ejemplo que está configurado para implementar las técnicas de esta divulgación. La Figura 3 se proporciona para propósitos de explicación y no es limitante de las técnicas según se ejemplifican y se describen de forma general en esta divulgación. Para fines de explicación, esta divulgación describe el decodificador 30 de vídeo en el contexto de la codificación de HEVC. Sin embargo, las técnicas de esta divulgación pueden ser aplicables a otros métodos o normas de codificación.

En el ejemplo de la Figura 3, el decodificador 30 de vídeo incluye una pluralidad de componentes funcionales. Los componentes funcionales del decodificador 30 de vídeo incluyen una unidad 150 de decodificación por entropía, una unidad 152 de procesamiento de predicción, una unidad 154 de cuantificación inversa, una unidad 156 de procesamiento de transformada inversa, una unidad 158 de reconstrucción, una unidad 159 de filtro y una memoria intermedia 160 de imágenes decodificadas. La unidad 152 de procesamiento de predicción incluye una unidad 162 de compensación de movimiento y una unidad 164 de procesamiento de intra-predicción. En otros ejemplos, el decodificador 30 de vídeo puede incluir más, menos o diferentes componentes funcionales.

El decodificador 30 de vídeo puede recibir un flujo de bits que comprende una pluralidad de elementos de sintaxis. La unidad 150 de decodificación por entropía puede analizar el flujo de bits para extraer elementos de sintaxis del flujo de bits. Como parte del análisis del flujo de bits, la unidad 150 de decodificación por entropía puede decodificar elementos de sintaxis codificados por entropía en el flujo de bits. La unidad 152 de procesamiento de predicción, la unidad 154 de cuantificación inversa, la unidad 156 de procesamiento de transformada inversa, la unidad 158 de reconstrucción y la unidad 159 de filtro pueden generar datos de vídeo decodificados basándose en los elementos de sintaxis extraídos del flujo de bits.

El flujo de bits puede comprender una serie de unidades de NAL. Las unidades de NAL del flujo de bits pueden incluir unidades de NAL de SPS, unidades de NAL de PPS, unidades de NAL de SEI y así sucesivamente. Como parte del análisis del flujo de bits, la unidad 150 de decodificación por entropía puede extraer y decodificar por entropía los SPS de unidades NAL de SPS, PSS de unidades NAL de PPS, mensajes de SEI de unidades de NAL de SEI, etc.

Además, las unidades de NAL del flujo de bits pueden incluir unidades de NAL de corte codificadas. Como parte del análisis del flujo de bits, la unidad 150 de decodificación por entropía puede extraer y decodificar por entropía cortes codificados de las unidades de NAL de corte codificadas. Cada uno de los cortes codificados puede incluir un encabezado de corte y datos de corte. El encabezamiento de corte puede contener elementos de sintaxis que pertenece a un corte. Los elementos de sintaxis en el encabezado de corte pueden incluir un elemento de sintaxis que identifica un PPS asociado con una imagen que contiene el corte. La unidad 150 de decodificación por entropía puede realizar operaciones de decodificación por entropía, tales como operaciones de decodificación de CABAC, en elementos de sintaxis en el encabezado de corte codificado para recuperar el encabezado de corte.

Como parte de la extracción de los datos del corte de las unidades de NAL de corte codificada, la unidad 150 de decodificación por entropía puede extraer elementos de sintaxis de las CU codificadas en los datos de corte. Los elementos de sintaxis extraídos pueden incluir elementos de sintaxis asociados con bloques de coeficientes. La unidad 150 de decodificación por entropía puede realizar operaciones de decodificación CABAC en algunos de los elementos de sintaxis.

Como parte de la realización de la decodificación CABAC en un elemento de sintaxis, la unidad 150 de decodificación por entropía puede identificar un contexto de codificación. La unidad 150 de decodificación por entropía puede dividir un intervalo en un subintervalo inferior y un subintervalo superior. Uno de los subintervalos puede estar asociado con el valor 0 y el otro subintervalo puede estar asociado con el valor 1. Las anchuras de los subintervalos pueden ser proporcionales a las probabilidades indicadas para los valores asociados por el contexto de codificación identificado. Si el valor codificado está dentro del subintervalo inferior, la unidad 150 de decodificación por entropía puede decodificar un binario que tiene el valor asociado con el subintervalo inferior. Si el valor codificado está dentro del subintervalo superior, la unidad 150 de decodificación por entropía puede decodificar un binario que tiene el valor asociado con el subintervalo superior. Para decodificar un binario siguiente del elemento de sintaxis, la unidad 150 de decodificación por entropía puede repetir estos pasos siendo el intervalo el subintervalo que contiene el valor codificado. Cuando la unidad 150 de decodificación por entropía repite estos pasos para el siguiente binario, la unidad 150 de decodificación por entropía puede usar probabilidades modificadas basándose en las probabilidades indicadas por el contexto de codificación identificado y los binarios decodificados. La unidad 150 de decodificación por entropía puede deshacer el proceso de conversión a binario de los binarios para recuperar el elemento de sintaxis.

Además, el decodificador 30 de vídeo puede realizar una operación de reconstrucción en una CU no particionada. Para realizar la operación de reconstrucción en una CU no particionada, el decodificador 30 de vídeo 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 vídeo puede reconstruir un bloque de píxeles residuales asociado con la CU.

Como parte de realizar una operación de reconstrucción en una TU de una CU, la unidad 154 de cuantificación inversa puede cuantificar inversamente, es decir, descuantificar, bloques de coeficientes asociados con la TU. La unidad 154 de cuantificación inversa puede usar un valor QP asociado con la CU de la TU para determinar un grado de cuantificación y, de igual forma, un grado de cuantificación inversa para que se aplique a la unidad 154 de cuantificación inversa.

Después de que la unidad 154 de cuantificación inversa cuantifique inversamente un bloque de coeficientes, la unidad 156 de procesamiento de transformada inversa puede aplicar una o más transformadas inversas al bloque de coeficientes para generar un bloque de muestra residual asociado con la TU. Por ejemplo, la unidad 156 de procesamiento de transformada inversa puede aplicar una DCT inversa, una transformada de números enteros inversa, una transformada de Karhunen-Loeve inversa (KLT), una transformada rotacional inversa, una transformada direccional inversa u otra transformada inversa al bloque de coeficientes.

Si se codifica una PU en modo de salto o la información de movimiento de la PU se codifica usando el modo de unión, la unidad 162 de compensación de movimiento puede generar una lista de candidatos de unión para la PU. La unidad 162 de compensación de movimiento puede identificar un candidato de unión seleccionado en la lista de candidatos de unión. La unidad 162 de compensación de movimiento puede generar bloques de muestra predictivos para la PU basándose en uno o más bloques de referencia asociados con la información de movimiento indicada por el candidato de unión seleccionado.

Si la información de movimiento de una PU se codifica usando el modo AMVP, la unidad 162 de compensación de movimiento puede generar una lista de candidatos de predictores de MV de 0 y/o una lista de candidatos de predictores de MV de lista 1. La unidad 162 de compensación de movimiento puede determinar un candidato de predictor de MV de la lista 0 seleccionado y/o un candidato de predictor de MV de la lista 1 seleccionado. A continuación, la unidad 162 de compensación de movimiento puede determinar un vector de movimiento de la lista 0 para la PU y/o un vector de movimiento de la lista 1 para la PU basándose en una diferencia de vector de movimiento (MVD) de la lista 0, una MVD de lista 1, un vector de movimiento de la lista 0 especificado por el candidato de predictor de MV de la lista 0 seleccionado, y/o un vector de movimiento de la lista 1 especificado por el candidato de predictor de MV de la lista 1 seleccionado. La unidad 162 de compensación de movimiento puede generar bloques de muestra predictivos para la PU basándose en bloques de referencia asociados con el vector de movimiento de la lista 0 y un índice de imagen de referencia de la lista 0 y/o un vector de movimiento de la lista 1 y un índice de imagen de referencia de la lista 1.

En algunos ejemplos, la unidad 162 de compensación de movimiento puede perfeccionar los bloques de muestra predictivos de una PU realizando una interpolación basándose en filtros de interpolación. Los identificadores de los filtros de interpolación que se utilizarán para la compensación de movimiento con precisión de subpíxeles pueden incluirse en los elementos de sintaxis. La unidad 162 de compensación de movimiento puede usar los mismos filtros de interpolación usados por el codificador 20 de vídeo durante la generación de los bloques de muestra predictivos de la PU para calcular valores interpolados para sub-muestras de números enteros de un bloque de referencia. La unidad 162 de compensación de movimiento puede determinar los filtros de interpolación usados por el codificador 20 de vídeo de acuerdo con la información de sintaxis recibida y usar los filtros de interpolación para producir los bloques de muestra predictivos.

Si una PU se codifica usando intra predicción, la unidad 164 de procesamiento de intra-predicción puede realizar intra predicción para generar bloques de muestra predictivos para la PU. Por ejemplo, la unidad 164 de procesamiento de intra-predicción puede determinar un modo de intra predicción para la PU basándose en elementos de sintaxis en el flujo de bits. La unidad 164 de procesamiento de intra-predicción puede usar el modo de intra predicción para generar datos predictivos (por ejemplo, muestras predictivas) para la PU basándose en los bloques de muestra de las PU espacialmente vecinas.

La unidad 158 de reconstrucción puede usar los bloques de muestra residuales asociados con las TU de una CU y los bloques de muestra predictiva de las PU de la CU, es decir, datos de intra-predicción o datos de inter-predicción, según sea aplicable, para reconstruir los bloques de muestra de la CU. En particular, la unidad 158 de reconstrucción puede añadir muestras de los bloques de muestra residuales a las muestras correspondientes de los bloques de muestra predictivos para reconstruir los bloques de muestra de la CU.

La unidad 159 de filtro puede realizar una operación de desbloqueo para reducir los artefactos de bloqueo asociados con la CU. El decodificador 30 de vídeo puede almacenar los bloques de muestra de la CU en la memoria intermedia 160 de imágenes decodificadas. La memoria intermedia 160 de imágenes decodificadas puede proporcionar imágenes de referencia para una compensación de movimiento posterior, intra predicción y presentación en un dispositivo de visualización, tal como el dispositivo 32 de visualización de la Figura 1. Por ejemplo, el decodificador 30 de vídeo puede realizar, basándose en los bloques de muestra en la memoria intermedia 160 de imágenes decodificadas, operaciones de intra predicción o inter predicción en PU de otras CU.

La Figura 4 es un diagrama de flujo que ilustra una operación 200 de ejemplo del codificador 20 de vídeo, de acuerdo con una o más técnicas de esta divulgación. En el ejemplo de la Figura 4, el codificador 20 de vídeo puede codificar particiones de imagen (por ejemplo, piezas, ondas, cortes de entropía, etc.) de un corte de datos (202) de vídeo. El

corte puede incluir múltiples unidades de codificación. Al codificar una partición de imágenes, el codificador 20 de vídeo genera una partición de imagen codificada. El codificador 20 de vídeo puede generar una unidad de NAL de corte codificada que incluye las particiones de imagen codificada del corte (204).

Además, el codificador 20 de vídeo puede generar un mensaje de SEI que indica los puntos de entrada de las particiones (206) de imagen codificada. Es decir, el mensaje de SEI puede indicar las posiciones en la unidad de NAL de corte codificada de las particiones de imagen codificada. El mensaje de SEI puede tener el formato descrito anteriormente. Por ejemplo, el codificador 20 de vídeo puede generar una unidad de NAL de SEI que incluye el mensaje de SEI.

El codificador 20 de vídeo puede generar un flujo de bits que incluye la unidad de NAL de corte codificada y el mensaje (208) de SEI. El flujo de bits incluye una representación codificada de datos de vídeo. El codificador 20 de vídeo puede emitir el flujo de bits de varias formas. Por ejemplo, el codificador 20 de vídeo puede transmitir el flujo de bits en una red de comunicaciones, tal como Internet. En otro ejemplo, el codificador 20 de vídeo puede emitir el flujo de bits a un medio de almacenamiento legible por ordenador, tal como una unidad de disco duro o una memoria.

La Figura 5 es un diagrama de flujo que ilustra una operación 250 de ejemplo del decodificador 30 de vídeo, de acuerdo con una o más técnicas de esta divulgación. En el ejemplo de la Figura 5, el decodificador 30 de vídeo puede recibir una unidad de NAL de corte codificada que contiene una pluralidad de particiones (252) de imagen codificada. Además, el decodificador 30 de vídeo puede recibir un mensaje de SEI, por ejemplo, en una unidad de NAL no de VCL, que indica puntos de entrada de las particiones de imagen codificada en la unidad (254) de NAL de corte codificad. El decodificador 30 de vídeo puede recibir la unidad de NAL de corte codificada y el mensaje de SEI de diversas formas. Por ejemplo, el decodificador 30 de vídeo puede recibir la unidad de NAL de corte codificada y el mensaje de SEI desde un servidor remoto a través de una red de comunicaciones, tal como Internet. En otro ejemplo, el decodificador 30 de vídeo puede recibir la unidad de NAL de corte codificada y el mensaje de SEI desde un medio de almacenamiento legible por ordenador, tal como una unidad de disco duro o una memoria.

El decodificador 30 de vídeo puede determinar, basándose al menos en parte en los puntos de entrada indicados en el mensaje de SEI, ubicaciones en la unidad de NAL de corte codificada de las particiones (256) de imagen codificada. Por ejemplo, el mensaje de SEI puede indicar que una partición de imagen codificada particular empieza en el byte de orden 256 de los datos de corte de la unidad de NAL de corte codificada. El decodificador 30 de vídeo puede decodificar las particiones (258) de imagen codificada. En algunos ejemplos, el decodificador 30 de vídeo puede decodificar dos o más de las particiones de imagen codificada en paralelo.

El decodificador 30 de vídeo puede emitir los bloques de píxeles decodificados de las particiones (260) de imagen codificada. En algunos ejemplos, el decodificador 30 de vídeo emite los bloques de píxeles decodificados para visualizarlos en el dispositivo 32 de visualización. En otro ejemplo, el decodificador 30 de vídeo puede emitir los bloques de píxeles decodificados a un codificador de vídeo. En este ejemplo, el codificador de vídeo puede volver a codificar los bloques de píxeles decodificados como parte de una operación de transcodificación de vídeo.

La Figura 6 es un diagrama de flujo que ilustra una operación 280 de ejemplo del codificador 20 de vídeo para generar una unidad de NAL de corte codificada que tiene un encabezado de corte alineado en bytes, de acuerdo con una o más técnicas de esta divulgación. En el ejemplo de la Figura 6, el codificador 20 de vídeo puede generar una pluralidad de particiones (282) de imagen codificada. Cada una de las particiones de imagen codificada está asociada con un conjunto diferente de CU en una imagen en los datos de vídeo. Además, el codificador 20 de vídeo puede generar una unidad de NAL que contiene un encabezado de corte y la pluralidad de particiones de imagen codificada (284). El encabezado de corte puede incluir uno o más bits de relleno que alinean el encabezado de corte con un límite de bytes.

La Figura 7 es un diagrama de flujo que ilustra una operación 300 de ejemplo del codificador 20 de vídeo para generar un encabezado de corte alineado en bytes, de acuerdo con una o más técnicas de esta divulgación. En el ejemplo de la Figura 7, el codificador 20 de vídeo puede generar un encabezado (302) de corte. Además, el codificador 20 de vídeo puede determinar si un elemento de sintaxis en un conjunto de parámetros (por ejemplo, un conjunto de parámetros de secuencia, un conjunto de parámetros de imagen, un conjunto de parámetros de adaptación, etc.) indica que el encabezado de corte debe alinearse con un límite (304) de bytes.

Si el elemento de sintaxis indica que el encabezado de corte debe alinearse con un límite de bytes ("SÍ" de 304), el codificador 20 de vídeo puede determinar si el encabezado de corte está alineado con un límite (306) de bytes. Si el encabezado de corte no está alineado con un límite de bytes ("NO" de 306), el codificador 20 de vídeo puede añadir un bit al encabezado (308) del corte. Después de añadir el bit al encabezado de corte, el codificador 20 de vídeo puede determinar de nuevo si el encabezado de corte está alineado con un límite (306) de bytes. De esta forma, el codificador 20 de vídeo puede continuar añadiendo bits al encabezado de corte hasta que el encabezado de corte esté alineado en bytes.

Si el encabezado de corte está alineado con un límite de bytes ("SÍ" de 306) o si el elemento de sintaxis indica que el encabezado de corte no necesita alinearse con un límite de bytes ("NO" de 304), la operación 300 está completa.

La Figura 8 es un diagrama de flujo que ilustra una operación 330 de ejemplo del decodificador 30 de vídeo en el que el decodificador 30 de vídeo recibe una unidad de NAL de corte codificada que tiene un encabezado de corte alineado en bytes, de acuerdo con una o más técnicas de esta divulgación. En el ejemplo de la Figura 8, el decodificador 30 de vídeo puede recibir una unidad (332) de NAL. La unidad de NAL puede contener un encabezado de corte y una pluralidad de particiones de imagen codificada de los datos de vídeo. El encabezado de corte puede incluir uno o más bits de relleno que alinean el encabezado de corte con un límite de bytes. El decodificador 30 de vídeo puede decodificar dos o más de las particiones de imagen codificada en paralelo (334).

La Figura 9 es un diagrama conceptual que ilustra el procesamiento paralelo de frente de onda. Como se describió anteriormente, una imagen puede dividirse en bloques de píxeles, cada uno de los cuales está asociado a un bloque de árbol. La Figura 9 ilustra los bloques de píxeles asociados con los bloques del árbol como una cuadrícula de cuadrados blancos. La imagen incluye las filas 350A-350E de bloques de árbol (colectivamente, "las filas 350 de bloques de árbol").

Un primer hilo puede codificar bloques de árbol en la fila 350A de bloques de árbol. Al mismo tiempo, otros hilos pueden codificar bloques de árbol en las filas 350B, 350C y 350D de bloques de árbol. En el ejemplo de la Figura 9, el primer hilo codifica actualmente un bloque 352A de árbol, un segundo hilo codifica actualmente un bloque 352B de árbol, un tercer hilo codifica actualmente un bloque 352C de árbol y un cuarto hilo codifica actualmente un bloque 352D de árbol. Esta descripción puede referirse a los bloques 352A, 352B, 352C y 352D de árbol colectivamente como "bloques 352 de árbol actuales". Debido a que el codificador de vídeo 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 352 de árbol actuales se desplazan horizontalmente entre sí por las anchuras de dos bloques de árbol.

En el ejemplo de la Figura 9, los hilos pueden usar datos de bloques de árbol indicados por las flechas grises gruesas para realizar intra predicción o inter predicción para CU en bloques 352 de árbol actuales. Los hilos también pueden utilizar datos de uno o más fotogramas de referencia para realizar inter predicción para las CU. Para codificar un bloque de árbol dado, un hilo puede seleccionar uno o más contextos CABAC basándose en la información asociada con bloques de árbol previamente codificados. El hilo puede usar uno o más contextos CABAC para realizar la codificación 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 hilo puede seleccionar el uno o más contextos CABAC basándose en 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 hilo puede seleccionar uno o más contextos CABAC basándose en 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 hilos pueden usar datos de las últimas CU de los bloques de árbol indicados por las flechas negras delgadas para seleccionar contextos CABAC para las primeras CU de los bloques 352 de árbol actuales.

La Figura 10 es un diagrama conceptual que ilustra un orden de codificación de bloques de árbol de ejemplo para una imagen 400 que está dividida en múltiples piezas 402A, 402B, 402C, 402D, 402E y 402F (colectivamente, las "piezas 402"). Cada bloque cuadrado de la imagen 400 representa un bloque de píxeles asociado con un bloque de árbol. Las líneas discontinuas gruesas indican los límites de las piezas de ejemplo. Los diferentes tipos de entramado cruzado corresponden a diferentes cortes.

Los números en los bloques de píxeles indican las posiciones de los bloques de árbol correspondientes (LCU) en un orden de codificación de pieza para la imagen 400. Como se ilustra en el ejemplo de la Figura 10, los bloques de árbol en la pieza 402A se codifican en primer lugar, seguidos de los bloques de árbol en la pieza 402B, seguidos de los bloques de árbol en la pieza 402C, seguidos de los bloques de árbol en la pieza 402D, seguidos de los bloques de árbol en la pieza 402E, seguidos de los bloques de árbol en la pieza 402F. Dentro de cada una de las piezas 402, los bloques de árbol se codifican según un orden de exploración por tramas.

Un codificador de vídeo puede generar cuatro unidades de NAL de corte codificadas para la imagen 400. La primera unidad de NAL de corte codificada puede incluir representaciones codificadas de bloques de árbol 1-18. Los datos de corte de la primera unidad de NAL de corte codificada pueden incluir dos particiones de imagen codificada. La primera partición de imagen codificada puede incluir las representaciones codificadas de los bloques de árbol 1-9. La segunda partición de imagen codificada puede incluir las representaciones codificadas de bloques de árbol 10-18. Por tanto, la primera unidad de NAL de corte codificada puede incluir una representación codificada de un corte que contiene múltiples piezas.

Una segunda unidad de NAL de corte codificada puede incluir representaciones codificadas de bloques de árbol 19-22. Los datos de corte de la segunda unidad de NAL de corte codificada pueden incluir una única partición de imagen codificada. Una tercera unidad de NAL de corte codificada puede incluir representaciones codificadas de bloques de árbol 23-27. Los datos de corte de la tercera unidad de NAL de corte codificada pueden incluir sólo una única partición de imagen codificada. Por tanto, la pieza 402C puede contener múltiples cortes. Una cuarta unidad de NAL de corte codificada puede incluir representaciones codificadas de bloques de árbol 28-45. Los datos de corte de la cuarta unidad de NAL de corte codificada pueden incluir tres particiones de imagen codificada, una para cada una de las

piezas 402D, 402E y 402F. Por tanto, la cuarta unidad de NAL de corte codificada puede incluir una representación codificada de un corte que contiene múltiples piezas.

La Figura 11 es un diagrama de flujo que ilustra una operación 430 de ejemplo del codificador 20 de vídeo para generar un flujo de bits que incluye valores de longitud variable que representan elementos de sintaxis asociados con desplazamientos de particiones de imagen codificada dentro de una unidad de NAL, de acuerdo con uno o más aspectos de esta divulgación. En el ejemplo de la Figura 11, el codificador 20 de vídeo puede codificar por entropía un primer elemento de sintaxis, un segundo elemento de sintaxis y una serie de elementos (432) de sintaxis de desplazamiento. Las longitudes de cada uno de los elementos de sintaxis de desplazamiento se pueden determinar basándose en el primer elemento de sintaxis. El número de elementos de sintaxis de desplazamiento en la serie de elementos de sintaxis de desplazamiento se puede determinar basándose en el segundo elemento de sintaxis. Las ubicaciones de una pluralidad de particiones de imagen codificada dentro de una unidad de NAL se pueden determinar basándose en los elementos de sintaxis de desplazamiento. El codificador 20 de vídeo puede generar un flujo de bits que incluye un valor de longitud variable que representa el primer elemento de sintaxis codificado por entropía, un valor de longitud variable que representa el segundo elemento de sintaxis codificado por entropía y valores de longitud fija que representan los elementos (434) de sintaxis de desplazamiento.

La Figura 12 es un diagrama de flujo que ilustra una operación 450 de ejemplo del decodificador de vídeo para decodificar datos de vídeo representados por un flujo de bits que incluye valores de longitud variable que representan elementos de sintaxis asociados con desplazamientos de particiones de imagen codificada en una unidad de NAL, de acuerdo con uno o más aspectos de esta divulgación. En el ejemplo de la Figura 12, el decodificador 30 de vídeo recibe un flujo de bits que incluye un valor de longitud variable codificado por entropía que representa un primer elemento de sintaxis, un valor de longitud variable codificado por entropía que representa un segundo elemento de sintaxis y una serie de elementos (452) de sintaxis de desplazamiento. Además, el decodificador 30 de vídeo puede decodificar por entropía el primer elemento de sintaxis y el segundo elemento (454) de sintaxis. El decodificador 30 de vídeo puede determinar, basándose en el primer elemento de sintaxis, las longitudes de cada uno de los elementos (456) de sintaxis de desplazamiento. El decodificador 30 de vídeo puede determinar, basándose en el segundo elemento de sintaxis, el número de elementos de sintaxis de desplazamiento en la serie de elementos (458) de sintaxis de desplazamiento.

Además, el decodificador 30 de vídeo puede analizar, basándose al menos en parte en las longitudes de cada uno de los elementos de sintaxis de desplazamiento y el número de elementos de sintaxis de desplazamiento en la serie de elementos de sintaxis de desplazamiento, los elementos (460) de sintaxis de desplazamiento. En algunos ejemplos, cuando el decodificador 30 de vídeo analiza los elementos de sintaxis de desplazamiento, el decodificador 30 de vídeo realiza una decodificación de longitud fija para cada uno de los elementos de sintaxis de desplazamiento. La decodificación de longitud fija también se puede denominar un modo especial de decodificación por entropía. De manera similar, la codificación de longitud fija también puede denominarse un modo especial de codificación por entropía.

El decodificador 30 de vídeo puede determinar, basándose en los elementos de sintaxis de desplazamiento, ubicaciones de una pluralidad de particiones de imagen codificada dentro de una unidad (462) de NAL. En algunos ejemplos, un encabezado de corte de la unidad de NAL incluye el primer elemento de sintaxis, el segundo elemento de sintaxis y la serie de elementos de sintaxis de desplazamiento. En otros ejemplos, un mensaje de SEI incluye el primer elemento de sintaxis, el segundo elemento de sintaxis y la serie de elementos de sintaxis de desplazamiento. Además, el decodificador 30 de vídeo puede decodificar las particiones (464) de imagen codificada.

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 en o transmitirse a través de, como una o más instrucciones o código, un medio legible por ordenador, y ejecutarse por una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, lo que corresponde a un medio tangible tal como unos medios de almacenamiento de datos o medios de comunicación que incluyen cualquier medio que facilite la transferencia de un programa informático desde 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) medios de almacenamiento legibles por ordenador tangibles que son no transitorios o (2) un medio de comunicación tal como una señal u onda portadora. El medio de almacenamiento de datos puede ser cualquier medio disponible que pueda accederse por uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código 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, dichos 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 se pueda utilizar para almacenar el código de programa deseado en forma de instrucciones o estructuras de datos y al que se puede acceder mediante un ordenador. Además, cualquier conexión se denomina apropiadamente un medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, servidor u otra fuente remota usando un cable coaxial, cable de fibra óptica, par trenzado, línea digital de abonado (DSL) o tecnologías inalámbricas tales como infrarrojos, radio y

microondas, entonces el cable coaxial, cable de fibra óptica, par trenzado, DSL o tecnologías inalámbricas tales como infrarrojos, radio y microondas se incluyen en la definición de medio. Sin embargo, debería entenderse que medios de almacenamiento legibles por ordenador y medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que, en su lugar, se refieren a medios de almacenamiento tangibles no transitorios. Disco magnético y disco óptico, como se usan en este documento, incluyen disco compacto (CD), disco láser, disco óptico, disco versátil digital (DVD), disco flexible y disco Blu-ray donde los discos magnéticos normalmente reproducen datos magnéticamente, mientras que los discos ópticos reproducen datos ópticamente con láseres. Las combinaciones de los anteriores también deben incluirse dentro del alcance de los medios legibles por ordenador.

Las instrucciones pueden ejecutarse por uno o más procesadores, como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), matrices lógicas programables en campo (FP-GA) u otros sistemas integrados o discretos equivalentes. circuitos lógicos. Por consiguiente, el término "procesador", como se usa en el presente documento puede hacer referencia a cualquiera de la estructura anterior o cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento puede proporcionarse dentro de módulos de hardware y/o software dedicados configurados para codificar y decodificar, o incorporarse en un códec combinado. También, las técnicas podrían implementarse totalmente en uno o más circuitos o elementos lógicos.

Las técnicas de esta divulgación pueden implementarse en una amplia diversidad de dispositivos o aparatos, incluyendo un microteléfono inalámbrico, un circuito integrado (CI) o un conjunto de CI (por ejemplo, un conjunto de chips). Se describen diversos componentes, módulos o unidades en esta divulgación para destacar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no requieren necesariamente su realización por diferentes unidades de hardware. En su lugar, como se ha descrito anteriormente, diversas unidades pueden combinarse en una unidad de hardware de códec o proporcionarse por una colección de unidades de hardware inter operativas, incluyendo uno o más procesadores, como se ha descrito anteriormente, en conjunto con software y/o firmware adecuado.

REIVINDICACIONES

1. Un método para generar un flujo de bits que incluye datos de vídeo codificados para una imagen, incluyendo la imagen una pluralidad de bloques y se divide en una pluralidad de particiones de imagen, en donde cada partición de imagen está asociada con un subconjunto no superpuesto de los bloques de manera que cada bloque está asociado con exactamente una de las particiones de imagen, y en donde cada partición de imagen de la pluralidad de particiones de imagen está alineada por bytes de modo que el primer bit de una partición de imagen codificada es el primer bit de un byte direccionable y el último bit de la partición de imagen codificada es el último bit de un byte direccionable, comprendiendo el método:

generar una unidad de capa de abstracción de red, NAL, codificada que incluye datos para una o más particiones de imagen codificada;
generar una serie de elementos de sintaxis de desplazamiento, indicando cada elemento de sintaxis de desplazamiento el primer bit de una respectiva de las una o más particiones de imagen codificada en la unidad de NAL;
codificar por entropía un primer elemento de sintaxis, en donde el primer elemento de sintaxis indica el número de bits de cada uno de los elementos de sintaxis de desplazamiento;
codificar por entropía un segundo elemento de sintaxis, en donde el segundo elemento de sintaxis indica el número de elementos de sintaxis de desplazamiento generados para la unidad de NAL;
generar un flujo de bits que incluye un valor de longitud variable que representa el primer elemento de sintaxis codificado por entropía, un valor de longitud variable que representa el segundo elemento de sintaxis codificado por entropía, valores de longitud fija que representan cada uno de los elementos de sintaxis de desplazamiento y la unidad de NAL codificada.

2. El método de la reivindicación 1, que comprende además generar un mensaje de información de mejora complementaria, SEI, que incluye el primer elemento de sintaxis, el segundo elemento de sintaxis y la serie de elementos de sintaxis de desplazamiento.

3. El método de la reivindicación 2, en donde el mensaje de SEI incluye un elemento de sintaxis que indica que el mensaje de SEI pertenece a un tipo de mensajes de SEI que indican puntos de entrada de particiones de imagen codificada en unidades de NAL de corte codificadas, comprendiendo el método además generar una unidad de NAL de SEI que contiene el mensaje de SEI.

4. Un método para decodificar datos de vídeo para una imagen, incluyendo la imagen una pluralidad de bloques y subdividiéndose en una pluralidad de particiones de imagen, en donde cada partición de imagen está asociada con un subconjunto no superpuesto de los bloques, de modo que cada bloque está asociado exactamente con una de las particiones de imagen, y en donde cada partición de imagen de la pluralidad de particiones de imagen está alineada en bytes de modo que el primer bit de una partición de imagen codificada es el primer bit de un byte direccionable y el último bit de la partición de imagen codificada es el último bit de un byte direccionable, comprendiendo el método:

recibir un flujo de bits que incluye una unidad de capa de abstracción de red, NAL, codificada que incluye datos para una o más particiones de imagen codificada, un valor de longitud variable codificado por entropía que representa un primer elemento de sintaxis, un valor de longitud variable codificado por entropía que representa un segundo elemento de sintaxis y una serie de elementos de sintaxis de desplazamiento de longitud fija;
decodificar por entropía el primer elemento de sintaxis y el segundo elemento de sintaxis;
determinar, basándose en el primer elemento de sintaxis, el número de bits para cada uno de los elementos de sintaxis de desplazamiento;
determinar, basándose en el segundo elemento de sintaxis, el número de elementos de sintaxis de desplazamiento en la serie de elementos de sintaxis de desplazamiento;
analizar, basándose en el número de bits de cada uno de los elementos de sintaxis de desplazamiento y el número de elementos de sintaxis de desplazamiento en la serie de elementos de sintaxis de desplazamiento, los elementos de sintaxis de desplazamiento;
determinar, basándose en los elementos de sintaxis de desplazamiento, ubicaciones de una pluralidad de particiones de imagen codificada dentro de la unidad de capa de abstracción de red, NAL, que incluye una o más particiones de imagen codificada; y
decodificar las particiones de imagen codificada.

5. El método de la reivindicación 1 o la reivindicación 4, en donde cada una de las particiones de imagen codificada incluye un grupo de bloques de árbol codificados dentro de la unidad de NAL que están asociados con un único corte de entropía, pieza u onda de procesamiento paralelo de frente de onda, WPP.

6. El método de la reivindicación 1 o la reivindicación 4, en donde la serie de elementos de sintaxis de desplazamiento indican desplazamientos de bytes de las particiones de imagen codificada con respecto a las particiones de imagen codificada anteriores dentro de la unidad de NAL.

7. El método de la reivindicación 4, en donde el flujo de bits incluye un mensaje de SEI que incluye el primer elemento de sintaxis, el segundo elemento de sintaxis y la serie de elementos de sintaxis de desplazamiento.

8. El método de la reivindicación 7, en donde el mensaje de SEI incluye un elemento de sintaxis que indica que el mensaje de SEI pertenece a un tipo de mensajes de SEI que indican puntos de entrada de particiones de imagen codificada en unidades de NAL de corte codificadas.

9. El método de la reivindicación 4, en donde el flujo de bits incluye una unidad de NAL de SEI que incluye el mensaje de SEI.

10. El método de la reivindicación 9,

en donde la unidad de NAL de SEI precede inmediatamente en el flujo de bits a la unidad de NAL que contiene la pluralidad de particiones de imagen codificada; y
en donde el método comprende además determinar, basándose al menos en parte en la unidad de NAL de SEI que precede inmediatamente en el flujo de bits a la unidad de NAL que contiene la pluralidad de particiones de imagen codificada, que el mensaje de SEI es aplicable a la unidad de NAL que contiene la pluralidad de particiones de imagen codificada.

11. El método de la reivindicación 4, que comprende además proporcionar las particiones de imagen codificada a diferentes núcleos de procesamiento para la decodificación en paralelo de las particiones de imagen codificada y decodificar las respectivas particiones de imagen codificada en los núcleos de procesamiento.

12. El método de la reivindicación 1 o la reivindicación 4, en donde las longitudes de cada uno de los elementos de sintaxis de desplazamiento son iguales a un valor del primer elemento de sintaxis más 10.

13. Un dispositivo informático para generar un flujo de bits que incluye datos de vídeo codificados para una imagen, incluyendo la imagen una pluralidad de bloques y subdividiéndose en una pluralidad de particiones de imagen, en donde cada partición de imagen está asociada con un subconjunto no superpuesto de los bloques de modo que cada bloque está asociado exactamente con una de las particiones de imagen, y en donde cada partición de imagen de la pluralidad de particiones de imagen está alineada en bytes de modo que el primer bit de una partición de imagen codificada es el primer bit de un byte direccionable y el último bit de la partición de imagen codificada es el último bit de un byte direccionable, comprendiendo el dispositivo:

medios para generar una unidad de capa de abstracción de red, NAL, codificada que incluye datos para una o más particiones de imagen codificada;
medios para generar una serie de elementos de sintaxis de desplazamiento, indicando cada elemento de sintaxis de desplazamiento el primer bit de una respectiva de las una o más particiones de imagen codificada en la unidad de NAL;
medios para codificar por entropía un primer elemento de sintaxis, en donde el primer elemento de sintaxis indica el número de bits de cada uno de los elementos de sintaxis de desplazamiento;
medios para codificar por entropía un segundo elemento de sintaxis, en donde el segundo elemento de sintaxis indica el número de elementos de sintaxis de desplazamiento generados para la unidad de NAL;
medios para codificar por entropía un primer elemento de sintaxis y un segundo elemento de sintaxis, y generar una serie de elementos de sintaxis de desplazamiento, en donde las longitudes de cada uno de los elementos de sintaxis de desplazamiento se pueden determinar basándose en el primer elemento de sintaxis, el número de elementos de sintaxis de desplazamiento en la serie de los elementos de sintaxis de desplazamiento se puede determinar basándose en el segundo elemento de sintaxis, y las ubicaciones de una pluralidad de particiones de imagen codificada dentro de una unidad de NAL se pueden determinar basándose en los elementos de sintaxis de desplazamiento; y
medios para generar un flujo de bits que incluye un valor de longitud variable que representa el primer elemento de sintaxis codificado por entropía, un valor de longitud variable que representa el segundo elemento de sintaxis codificado por entropía, valores de longitud fija que representan los elementos de sintaxis de desplazamiento y la unidad de NAL codificada.

14. Un dispositivo informático para decodificar datos de vídeo para una imagen, incluyendo la imagen una pluralidad de bloques y subdividiéndose en una pluralidad de particiones de imagen, en donde cada partición de imagen está asociada con un subconjunto no superpuesto de los bloques, de modo que cada bloque está asociado con exactamente una de las particiones de imagen, y en donde cada partición de imagen de la pluralidad de particiones de imagen está alineada en bytes de modo que el primer bit de una partición de imagen codificada es el primer bit de un byte direccionable y el último bit de la partición de imagen codificada es el último bit de un byte direccionable, comprendiendo el dispositivo:

medios para recibir un flujo de bits que incluye una unidad de capa de abstracción de red, NAL, codificada que incluye datos para una o más particiones de imagen codificada, un valor de longitud variable codificado por entropía que representa un primer elemento de sintaxis, un valor de longitud variable codificado por entropía que representa un segundo elemento de sintaxis y una serie de elementos de sintaxis de desplazamiento;

medios para decodificar por entropía el primer elemento de sintaxis y el segundo elemento de sintaxis;
medios para determinar, basándose en el primer elemento de sintaxis, el número de bits de cada uno de los elementos de sintaxis de desplazamiento;
medios para determinar, basándose en el segundo elemento de sintaxis, el número de elementos de sintaxis de desplazamiento en la serie de elementos de sintaxis de desplazamiento;
medios para analizar, basándose en el número de bits de cada uno de los elementos de sintaxis de desplazamiento y el número de elementos de sintaxis de desplazamiento en la serie de elementos de sintaxis de desplazamiento, los elementos de sintaxis de desplazamiento;
medios para determinar, basándose en los elementos de sintaxis de desplazamiento, ubicaciones de una pluralidad de particiones de imagen codificada dentro de la unidad de capa de abstracción de red, NAL, que incluye una o más particiones de imagen codificada; y
medios para decodificar las particiones de imagen codificada.

15. Un medio de almacenamiento legible por ordenador que almacena instrucciones que, cuando se ejecutan por uno o más procesadores de un dispositivo informático, configuran el dispositivo informático para realizar el método de una cualquiera de las reivindicaciones 1 a 12.

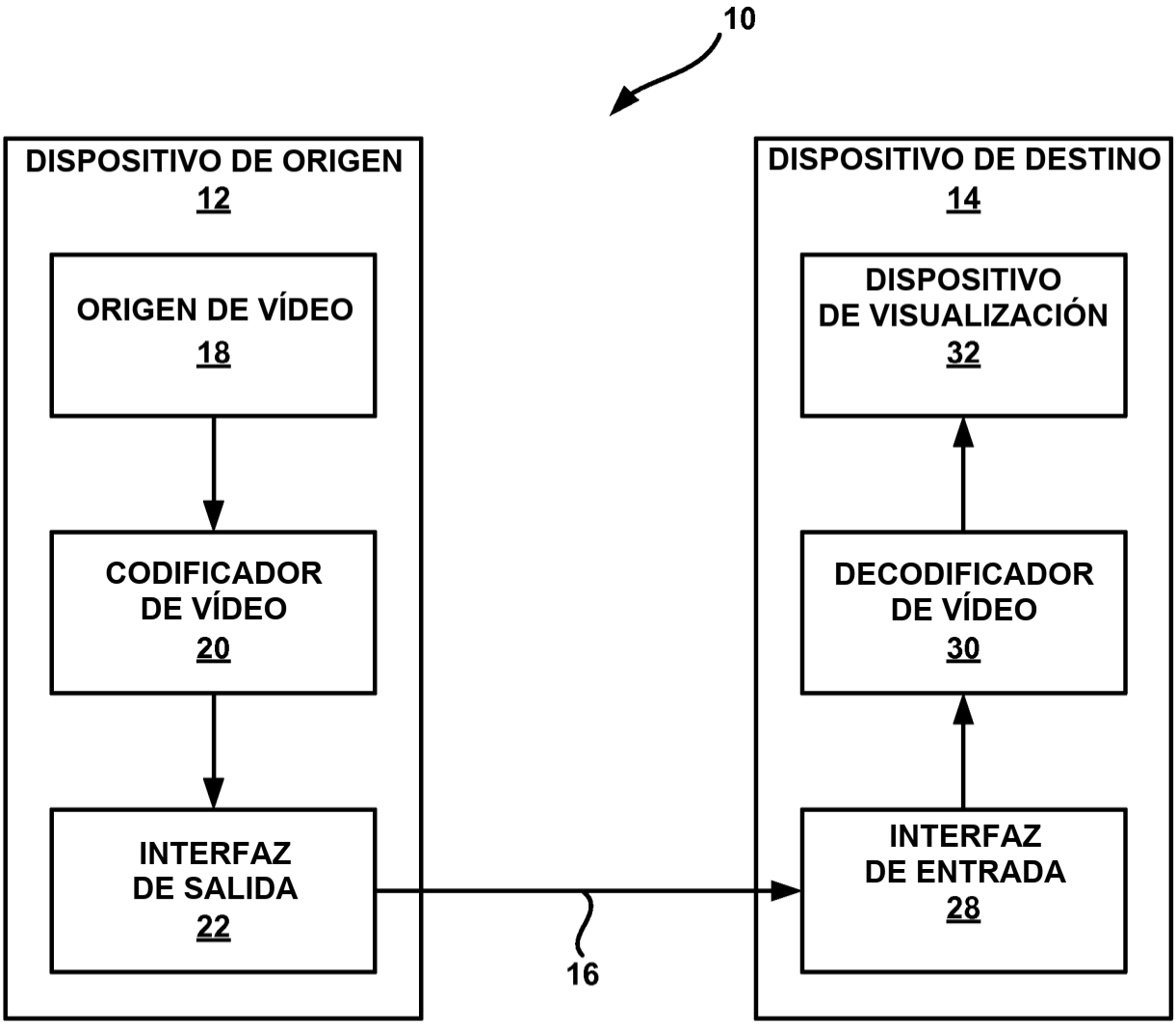

10
DISPOSITIVO DE ORIGEN
12
ORIGEN DE VÍDEO
18
CODIFICADOR
DE VÍDEO
20
INTERFAZ
DE SALIDA
22
16
DISPOSITIVO DE DESTINO
14
DISPOSITIVO
DE VISUALIZACIÓN
32
DECODIFICADOR
DE VÍDEO
30
INTERFAZ
DE ENTRADA
28

FIG. 1

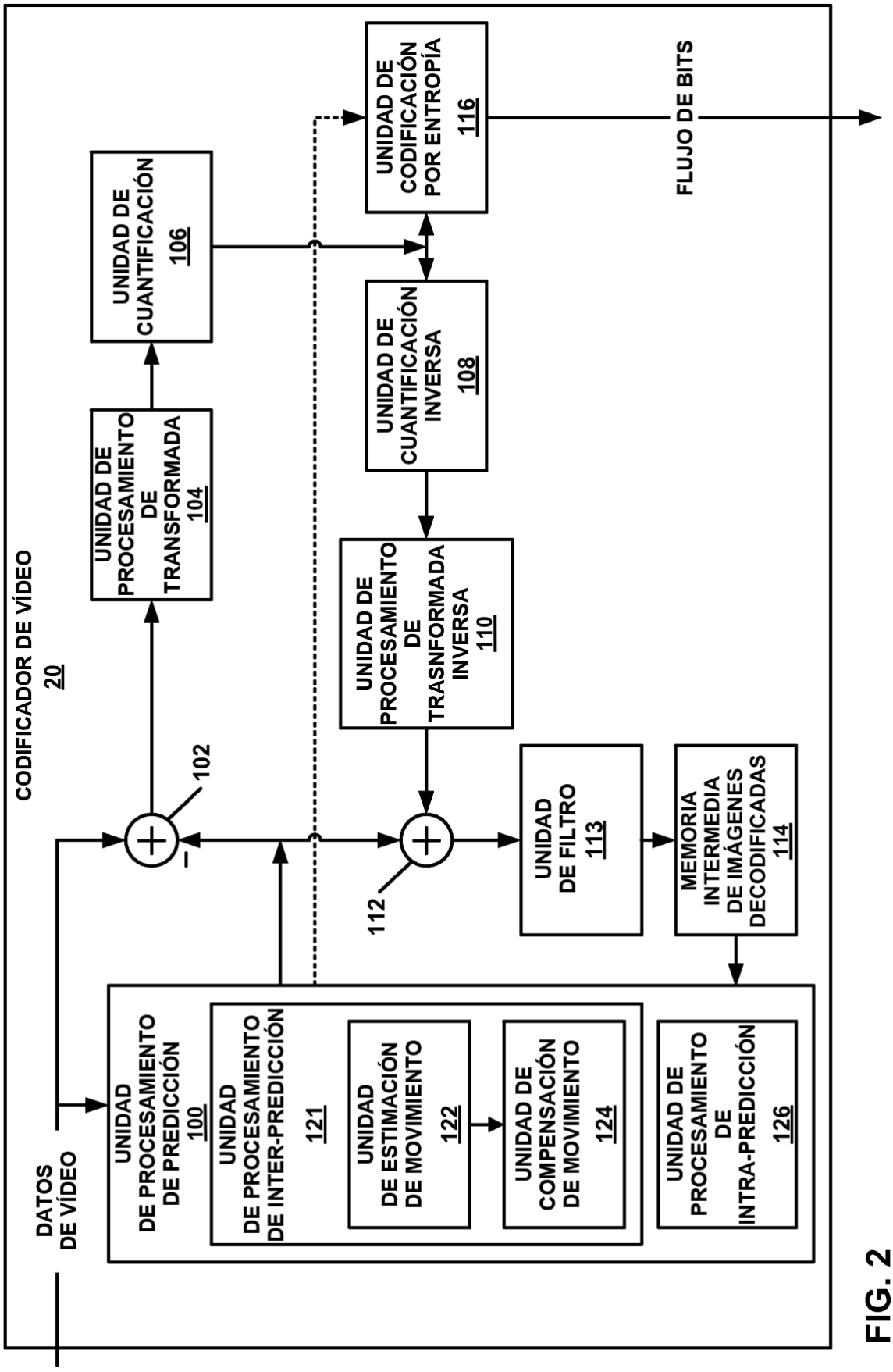

CODIFICADOR DE VÍDEO
20
DATOS DE VÍDEO
UNIDAD DE PROCESAMIENTO DE PREDICCIÓN
100
UNIDAD DE PROCESAMIENTO DE INTER-PREDICCIÓN
121
UNIDAD DE ESTIMACIÓN DE MOVIMIENTO
122
UNIDAD DE COMPENSACIÓN DE MOVIMIENTO
124
UNIDAD DE PROCESAMIENTO DE INTRA-PREDICCIÓN
126
102
UNIDAD DE PROCESAMIENTO DE TRANSFORMADA
104
UNIDAD DE CUANTIFICACIÓN
106
UNIDAD DE CUANTIFICACIÓN INVERSA
108
UNIDAD DE PROCESAMIENTO DE TRASNFORMADA INVERSA
110
112
UNIDAD DE FILTRO
113
MEMORIA INTERMEDIA DE IMÁGENES DECODIFICADAS
114
UNIDAD DE CODIFICACIÓN POR ENTROPÍA
116
FLUJO DE BITS

FIG. 2

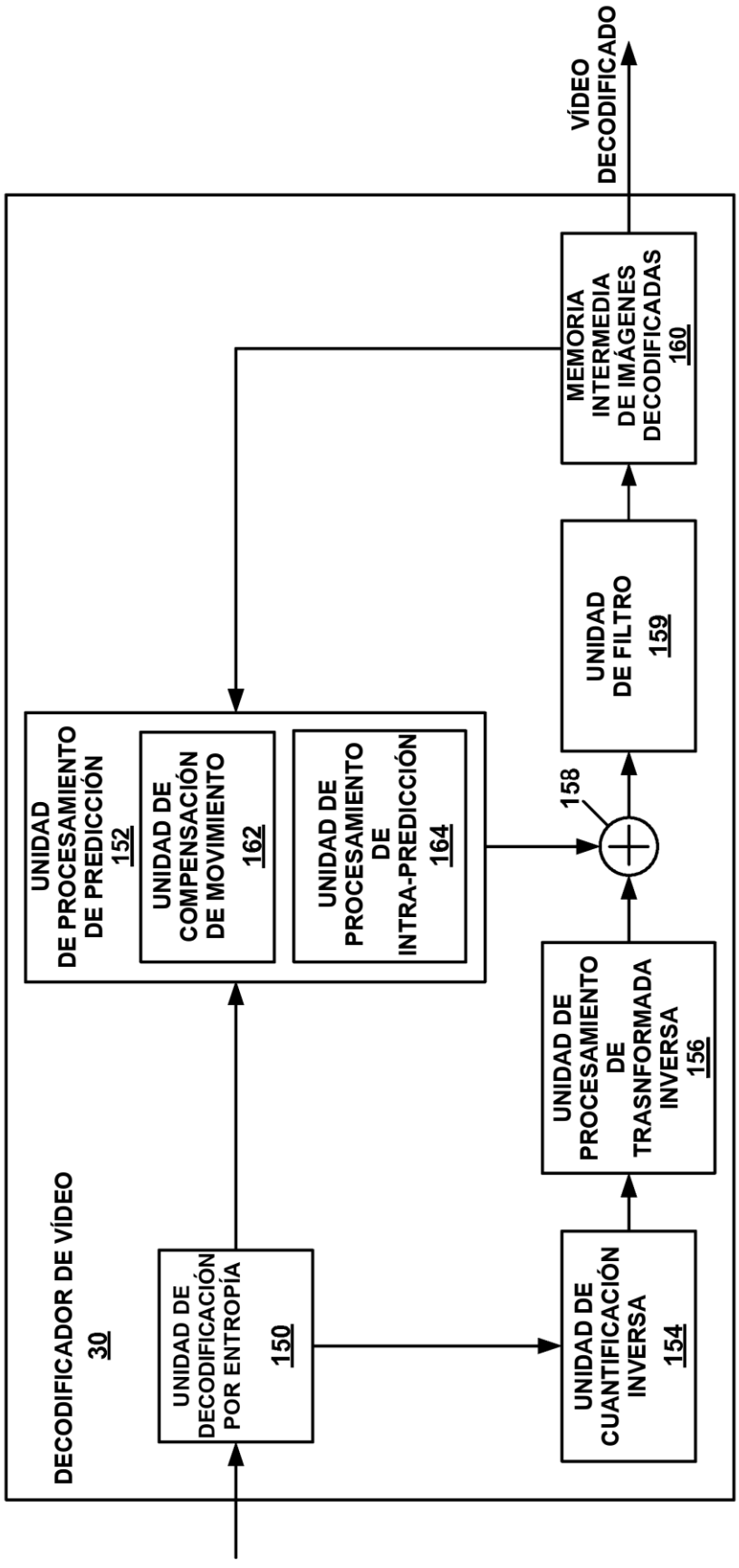

DECODIFICADOR DE VÍDEO
30
UNIDAD DE DECODIFICACIÓN POR ENTROPÍA
150
UNIDAD DE PROCESAMIENTO DE PREDICCIÓN
152
UNIDAD DE COMPENSACIÓN DE MOVIMIENTO
162
UNIDAD DE PROCESAMIENTO DE INTRA-PREDICCIÓN
164
UNIDAD DE CUANTIFICACIÓN INVERSA
154
UNIDAD DE PROCESAMIENTO DE TRASNFORMADA INVERSA
156
158
UNIDAD DE FILTRO
159
MEMORIA INTERMEDIA DE IMÁGENES DECODIFICADAS
160
VÍDEO DECODIFICADO

FIG. 3

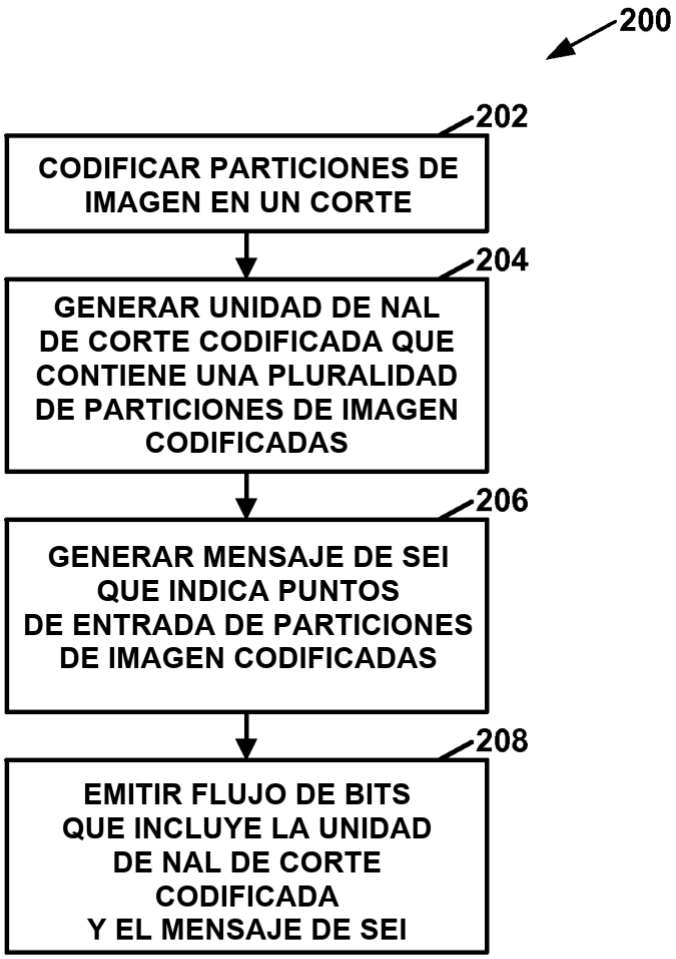

200
202
CODIFICAR PARTICIONES DE IMAGEN EN UN CORTE
204
GENERAR UNIDAD DE NAL DE CORTE CODIFICADA QUE CONTIENE UNA PLURALIDAD DE PARTICIONES DE IMAGEN CODIFICADAS
206
GENERAR MENSAJE DE SEI QUE INDICA PUNTOS DE ENTRADA DE PARTICIONES DE IMAGEN CODIFICADAS
208
EMITIR FLUJO DE BITS QUE INCLUYE LA UNIDAD DE NAL DE CORTE CODIFICADA Y EL MENSAJE DE SEI

FIG. 4

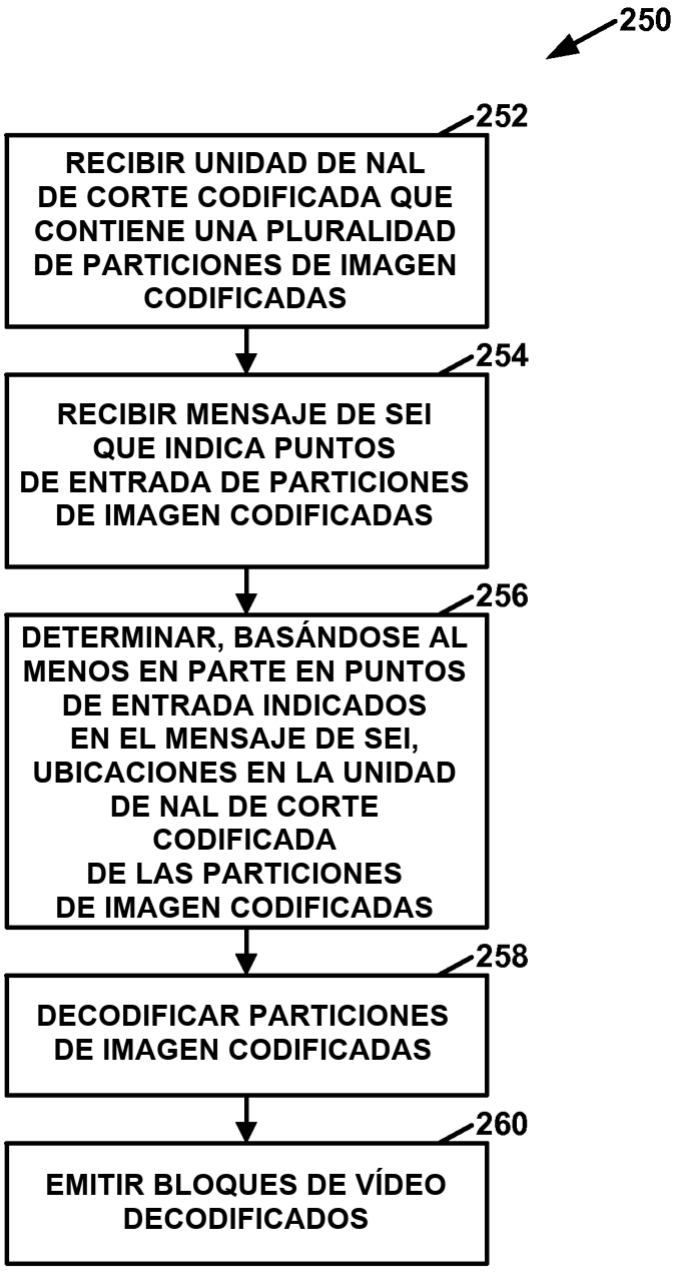

250
252
RECIBIR UNIDAD DE NAL DE CORTE CODIFICADA QUE CONTIENE UNA PLURALIDAD DE PARTICIONES DE IMAGEN CODIFICADAS
254
RECIBIR MENSAJE DE SEI QUE INDICA PUNTOS DE ENTRADA DE PARTICIONES DE IMAGEN CODIFICADAS
256
DETERMINAR, BASÁNDOSE AL MENOS EN PARTE EN PUNTOS DE ENTRADA INDICADOS EN EL MENSAJE DE SEI, UBICACIONES EN LA UNIDAD DE NAL DE CORTE CODIFICADA DE LAS PARTICIONES DE IMAGEN CODIFICADAS
258
DECODIFICAR PARTICIONES DE IMAGEN CODIFICADAS
260
EMITIR BLOQUES DE VÍDEO DECODIFICADOS

FIG. 5

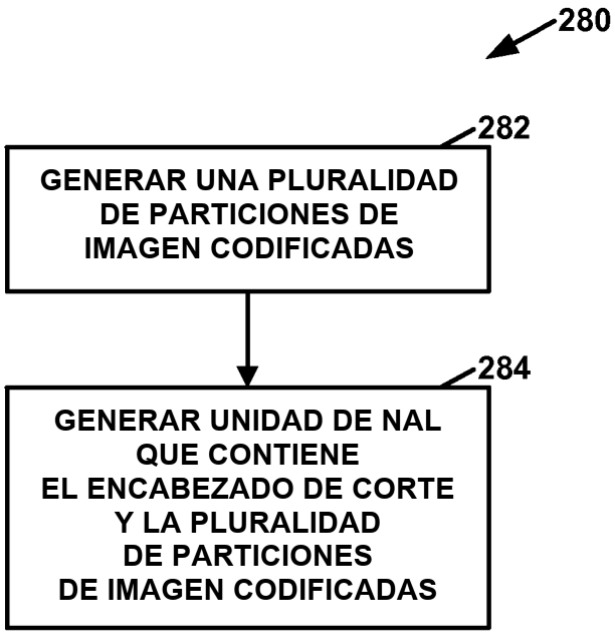

280
282
GENERAR UNA PLURALIDAD DE PARTICIONES DE IMAGEN CODIFICADAS
284
GENERAR UNIDAD DE NAL QUE CONTIENE EL ENCABEZADO DE CORTE Y LA PLURALIDAD DE PARTICIONES DE IMAGEN CODIFICADAS

FIG. 6

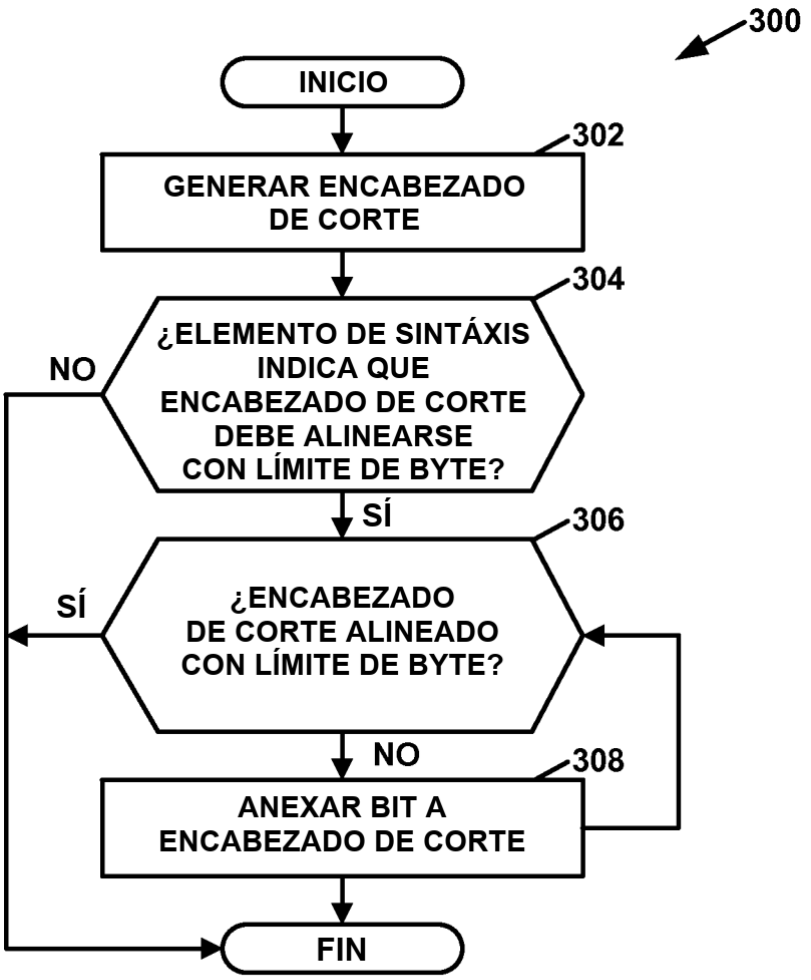

300
INICIO
302
GENERAR ENCABEZADO DE CORTE
304
¿ELEMENTO DE SINTÁXIS INDICA QUE ENCABEZADO DE CORTE DEBE ALINEARSE CON LÍMITE DE BYTE?
NO
SÍ
306
¿ENCABEZADO DE CORTE ALINEADO CON LÍMITE DE BYTE?
SÍ
NO
308
ANEXAR BIT A ENCABEZADO DE CORTE
FIN

FIG. 7

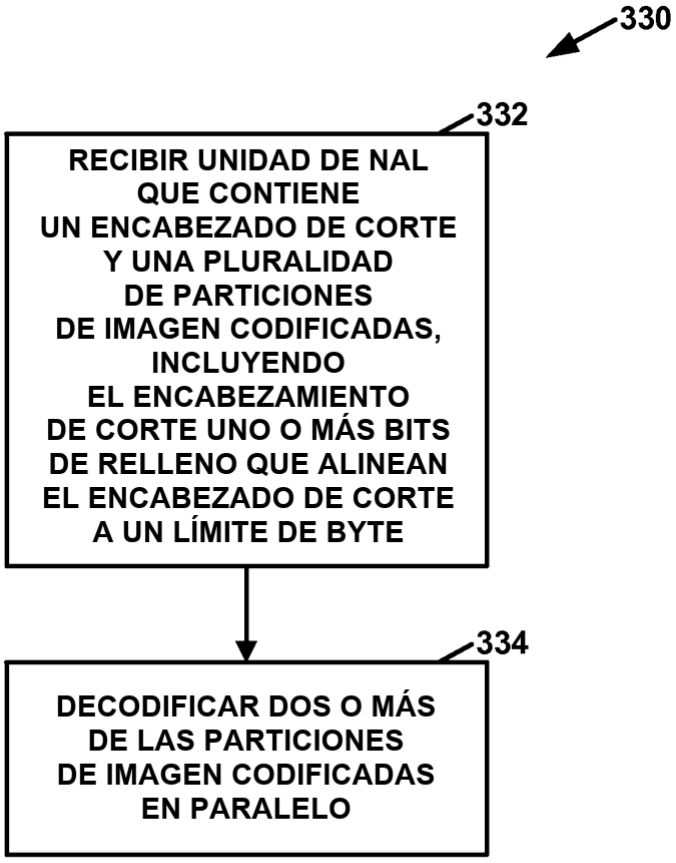

330
332
RECIBIR UNIDAD DE NAL QUE CONTIENE UN ENCABEZADO DE CORTE Y UNA PLURALIDAD DE PARTICIONES DE IMAGEN CODIFICADAS, INCLUYENDO EL ENCABEZAMIENTO DE CORTE UNO O MÁS BITS DE RELLENO QUE ALINEAN EL ENCABEZADO DE CORTE A UN LÍMITE DE BYTE
334
DECODIFICAR DOS O MÁS DE LAS PARTICIONES DE IMAGEN CODIFICADAS EN PARALELO

FIG. 8

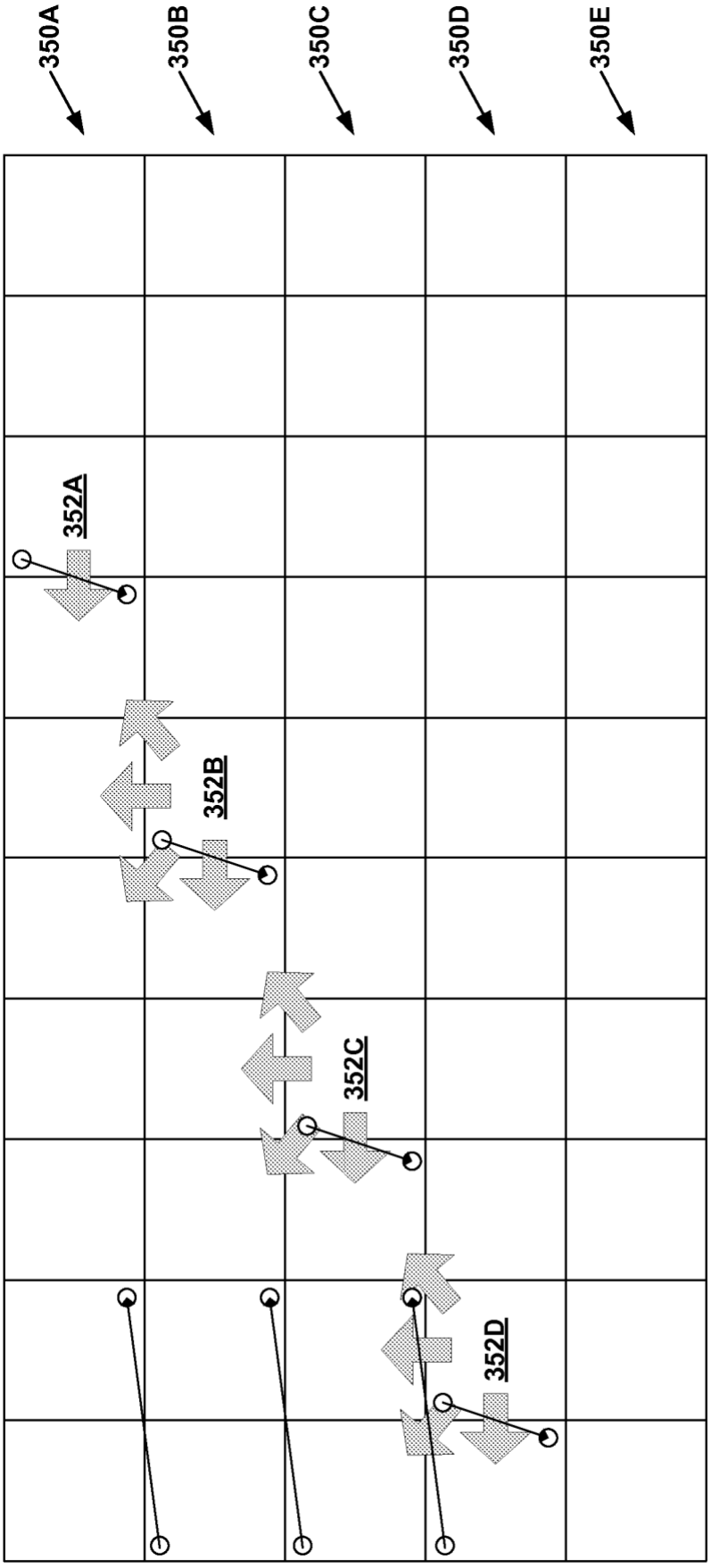


FIG. 9

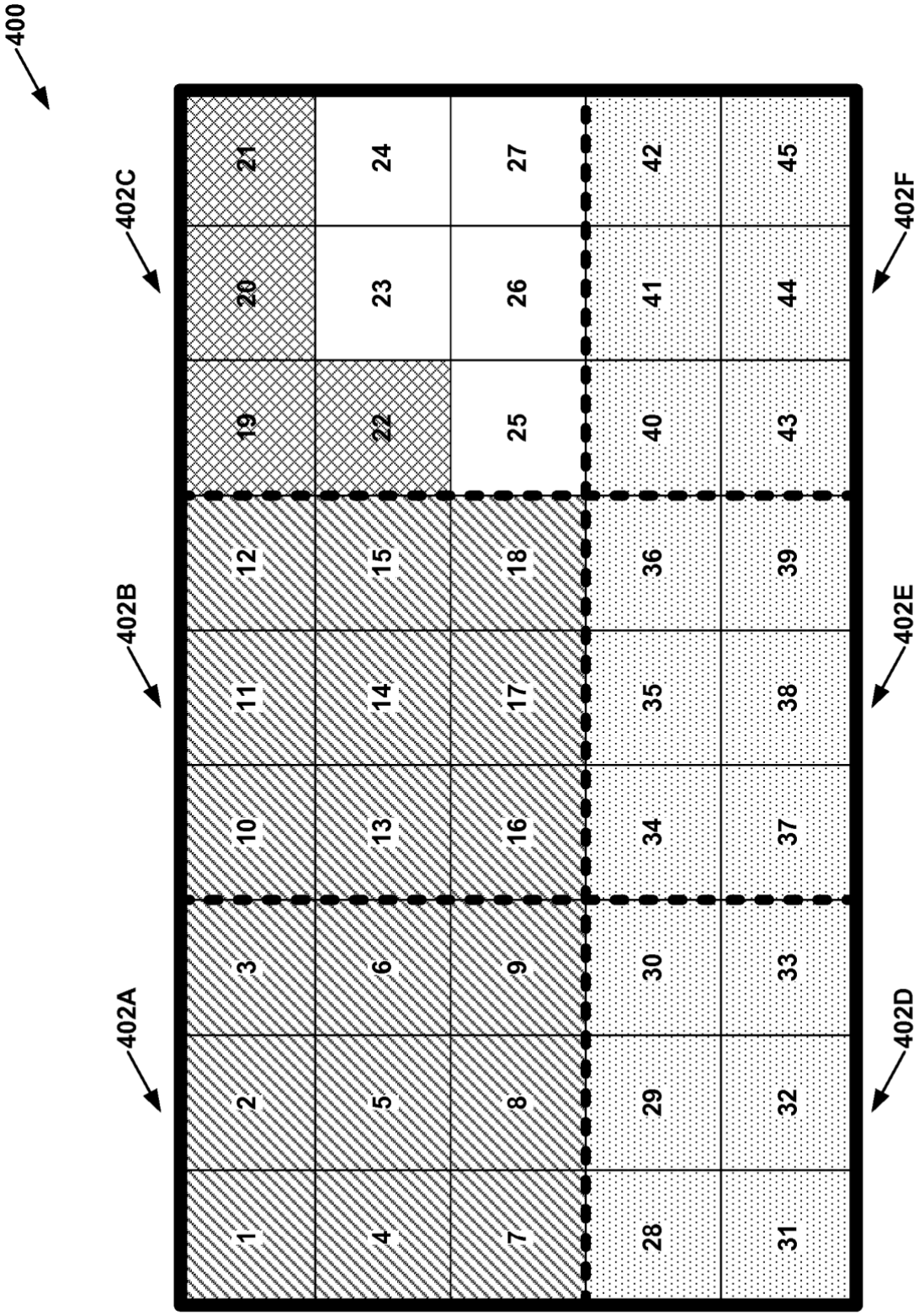

400
402A
402B
402C
402D
402E
402F
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45

FIG. 10

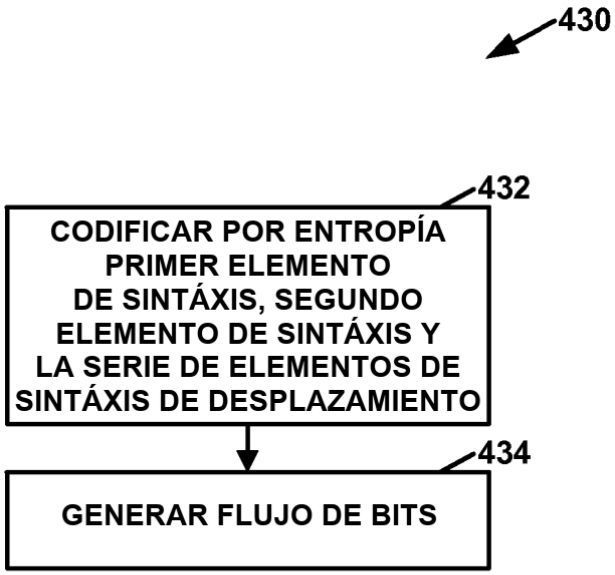

430
432
CODIFICAR POR ENTROPÍA PRIMER ELEMENTO DE SINTÁXIS, SEGUNDO ELEMENTO DE SINTÁXIS Y LA SERIE DE ELEMENTOS DE SINTÁXIS DE DESPLAZAMIENTO
434
GENERAR FLUJO DE BITS

FIG. 11

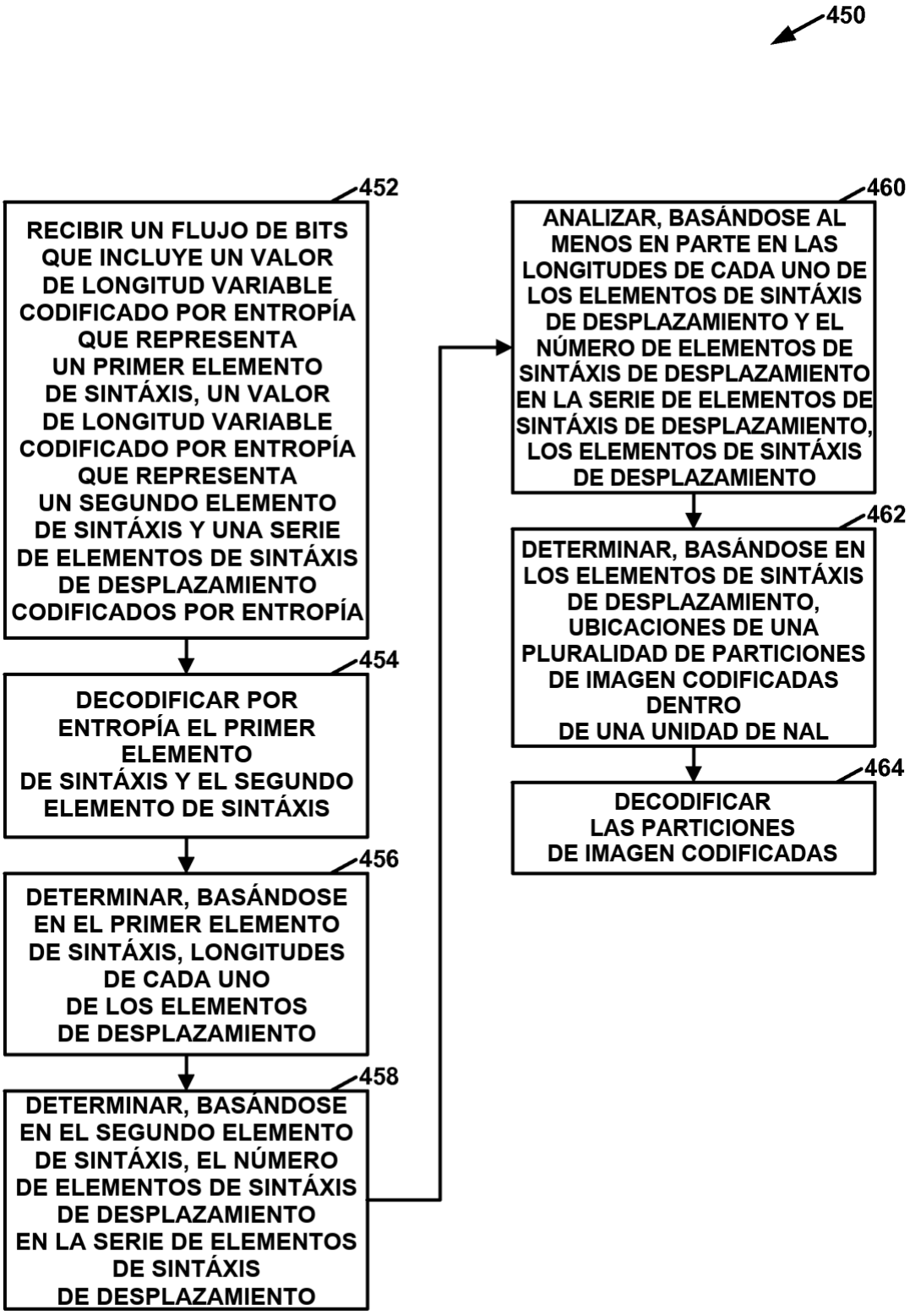

450
452
RECIBIR UN FLUJO DE BITS QUE INCLUYE UN VALOR DE LONGITUD VARIABLE CODIFICADO POR ENTROPÍA QUE REPRESENTA UN PRIMER ELEMENTO DE SINTÁXIS, UN VALOR DE LONGITUD VARIABLE CODIFICADO POR ENTROPÍA QUE REPRESENTA UN SEGUNDO ELEMENTO DE SINTÁXIS Y UNA SERIE DE ELEMENTOS DE SINTÁXIS DE DESPLAZAMIENTO CODIFICADOS POR ENTROPÍA
454
DECODIFICAR POR ENTROPÍA EL PRIMER ELEMENTO DE SINTÁXIS Y EL SEGUNDO ELEMENTO DE SINTÁXIS
456
DETERMINAR, BASÁNDOSE EN EL PRIMER ELEMENTO DE SINTÁXIS, LONGITUDES DE CADA UNO DE LOS ELEMENTOS DE DESPLAZAMIENTO
458
DETERMINAR, BASÁNDOSE EN EL SEGUNDO ELEMENTO DE SINTÁXIS, EL NÚMERO DE ELEMENTOS DE SINTÁXIS DE DESPLAZAMIENTO EN LA SERIE DE ELEMENTOS DE SINTÁXIS DE DESPLAZAMIENTO
460
ANALIZAR, BASÁNDOSE AL MENOS EN PARTE EN LAS LONGITUDES DE CADA UNO DE LOS ELEMENTOS DE SINTÁXIS DE DESPLAZAMIENTO Y EL NÚMERO DE ELEMENTOS DE SINTÁXIS DE DESPLAZAMIENTO EN LA SERIE DE ELEMENTOS DE SINTÁXIS DE DESPLAZAMIENTO, LOS ELEMENTOS DE SINTÁXIS DE DESPLAZAMIENTO
462
DETERMINAR, BASÁNDOSE EN LOS ELEMENTOS DE SINTÁXIS DE DESPLAZAMIENTO, UBICACIONES DE UNA PLURALIDAD DE PARTICIONES DE IMAGEN CODIFICADAS DENTRO DE UNA UNIDAD DE NAL
464
DECODIFICAR LAS PARTICIONES DE IMAGEN CODIFICADAS

FIG. 12