

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 998 471**

51 Int. Cl.:

**H04N 19/105** (2014.01)  
**H04N 19/50** (2014.01)  
**H04N 19/176** (2014.01)  
**H04N 19/119** (2014.01)  
**H04N 19/172** (2014.01)  
**H04N 19/136** (2014.01)

12

## TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **13.01.2017** **PCT/US2017/013485**  
87 Fecha y número de publicación internacional: **20.07.2017** **WO17123980**  
96 Fecha de presentación y número de la solicitud europea: **13.01.2017** **E 17703841 (1)**  
97 Fecha y número de publicación de la concesión europea: **18.12.2024** **EP 3403405**

54 Título: **Marco de árboles de múltiples tipos para la codificación de vídeo**

30 Prioridad:

**15.01.2016 US 201662279233 P**  
**12.01.2017 US 201715404634**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:  
**20.02.2025**

73 Titular/es:

**QUALCOMM INCORPORATED (100.00%)**  
**International IP Administration 5775 Morehouse**  
**Drive**  
**San Diego, CA 92121-1714, US**

72 Inventor/es:

**LI, XIANG;**  
**ZHANG, LI;**  
**CHIEN, WEI-JUNG;**  
**CHEN, JIANLE;**  
**ZHAO, XIN y**  
**KARCZEWICZ, MARTA**

74 Agente/Representante:

**ISERN JARA, Jorge**

ES 2 998 471 T3

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

## DESCRIPCIÓN

Marco de árboles de múltiples tipos para la codificación de vídeo

5 Esta solicitud reivindica el beneficio de la Solicitud Provisional U.S. No. 62/279,233, presentada el 15 de enero de 2016.

Campo técnico

10 Esta divulgación se refiere a la codificación de vídeo y a la decodificación de vídeo.

Antecedentes

15 Las capacidades de vídeo digital se pueden incorporar en un amplio rango de dispositivos, incluidos televisores digitales, sistemas de transmisión directa digital, sistemas de transmisión inalámbrica, asistentes digitales personales (PDAs), 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, teléfonos de radio celulares o satelitales, los llamados "teléfonos inteligentes", dispositivos de videoteleconferencia, dispositivos de transmisión continua de vídeo y similares. Los dispositivos de vídeo digital implementan técnicas de codificación 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), el estándar de Codificación de Vídeo de Alta Eficiencia (HEVC) y extensiones de dichos estándares. Los dispositivos de vídeo pueden transmitir, recibir, codificar, decodificar y/o almacenar información de vídeo digital de manera más eficiente utilizando la implementación de dichas técnicas de codificación de vídeo.

25 Las técnicas de codificación de vídeo incluyen la predicción espacial (intraimagen) y/o predicción temporal (interimagen) para reducir o eliminar la redundancia inherente a las secuencias de vídeo. Para la codificación de vídeo basada en bloques, un segmento de vídeo (por ejemplo, una imagen/fotograma de vídeo o una porción de una imagen de vídeo) puede partitionarse en bloques de vídeo, que también pueden denominarse bloques de árbol, unidades de codificación (CUs) y/o nodos de codificación. Las imágenes pueden denominarse fotogramas. Las imágenes de referencia pueden denominarse fotogramas de referencia.

30 La predicción espacial o temporal da como resultado un bloque predictivo para que se codifique un bloque. Los datos residuales representan las diferencias de píxeles entre el bloque original que se va a codificar y el bloque predictivo. Para una mayor compresión, los datos residuales se pueden transformar del dominio de píxeles a un dominio de transformación, lo que da como resultado coeficientes de transformación residuales, que luego se pueden cuantizar. Se puede aplicar la codificación de entropía para lograr aún más compresión. En el documento de patente JP2013229674A, se proporciona un dispositivo de codificación de imágenes que divide una imagen en unidades de predicción, genera una imagen de predicción para cada unidad de predicción y codifica la imagen, y la forma de cada una de las unidades de predicción se selecciona de una pluralidad de formas predeterminadas.

Resumen

45 Esta divulgación describe técnicas para particionar bloques de datos de vídeo utilizando un marco de árbol de múltiples tipos (MTT). La invención se define en las reivindicaciones independientes adjuntas. Las características opcionales se establecen en las reivindicaciones dependientes.

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

Breve descripción de los dibujos

55 La FIG. 1 es un diagrama de bloques que ilustra un ejemplo de un sistema de codificación y decodificación de vídeo configurado para implementar las técnicas de la divulgación.

La FIG. 2 es un diagrama conceptual que ilustra la estructura de la unidad de codificación (CU) en la Codificación de Vídeo de Alta Eficiencia (HEVC).

60 La FIG. 3 es un diagrama conceptual que ilustra ejemplos de tipos de partición para un modo de interpredicción.

La FIG. 4A es un diagrama conceptual que ilustra un ejemplo de partición de bloques utilizando una estructura de árbol cuaternario binario (QTBT).

65 La FIG. 4B es un diagrama conceptual que ilustra un ejemplo de estructura de árbol correspondiente a la partición de bloques utilizando la estructura QTBT de la FIG. 4A.

La FIG. 5A es un diagrama conceptual que ilustra ejemplos de tipos de particiones horizontales de árbol terciario.

5 La FIG. 5B es un diagrama conceptual que ilustra ejemplos de tipos de particiones horizontales de árbol terciario.

La FIG. 6A es un diagrama conceptual que ilustra la partición de árbol cuaternario.

10 La FIG. 6B es un diagrama conceptual que ilustra la partición vertical de árbol binario.

La FIG. 6C es un diagrama conceptual que ilustra la partición horizontal de árbol binario.

La FIG. 6D es un diagrama conceptual que ilustra la partición vertical de árbol del lado central.

15 La FIG. 6E es un diagrama conceptual que ilustra la partición horizontal de árbol del lado central.

La FIG. 7 es un diagrama conceptual que ilustra un ejemplo de partición de unidades de árbol de codificación (CTU) de acuerdo con las técnicas de esta divulgación.

20 La FIG. 8 es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo.

La FIG. 9 es un diagrama de bloques que ilustra un ejemplo de un decodificador de vídeo.

25 La FIG. 10A es un diagrama de flujo que ilustra un ejemplo de funcionamiento de un codificador de vídeo, de acuerdo con una técnica de esta divulgación.

La FIG. 10B es un diagrama de flujo que ilustra un ejemplo de funcionamiento de un decodificador de vídeo, de acuerdo con una técnica de esta divulgación.

30 La FIG. 11 es un diagrama de flujo que ilustra un ejemplo de funcionamiento de un codificador de vídeo, de acuerdo con otra técnica de ejemplo de esta divulgación.

35 La FIG. 12 es un diagrama de flujo que ilustra un ejemplo de funcionamiento de un decodificador de vídeo, de acuerdo con otra técnica de ejemplo de esta divulgación.

#### Descripción detallada

40 Esta divulgación está relacionada con la partición y/u organización de bloques de datos de vídeo (por ejemplo, unidades de codificación) en la codificación de vídeo basada en bloques. Las técnicas de esta divulgación pueden aplicarse en los estándares de codificación de vídeo. En diversos ejemplos que se describen a continuación, las técnicas de esta divulgación incluyen la partición de bloques de datos de vídeo utilizando tres o más estructuras de partición diferentes. En algunos ejemplos, se pueden utilizar tres o más estructuras de partición diferentes en cada profundidad de una estructura de árbol de codificación. Estas técnicas de partición

45 pueden denominarse partición de árbol de múltiples tipos (MTT). Utilizando las particiones MTT, los datos de vídeo pueden ser particionados más flexiblemente, por lo tanto permite una mayor eficiencia de codificación.

La FIG. 1 es un diagrama de bloques que ilustra un ejemplo del sistema de codificación y decodificación de vídeo 10 que puede utilizar técnicas de esta divulgación para particionar bloques de datos de vídeo, señalar y analizar tipos de particiones, y aplicar transformaciones y particiones de transformación adicionales. Como se muestra en la FIG. 1, el sistema 10 incluye un dispositivo fuente 12 que proporciona datos de vídeo codificados para ser decodificados posteriormente por un dispositivo destino 14. En particular, el dispositivo fuente 12 proporciona los datos de vídeo al dispositivo destino 14 a través de un medio legible por ordenador 16. El dispositivo fuente 12 y el dispositivo destino 14 pueden comprender cualquiera de un amplio rango de dispositivos, incluidos ordenadores de sobremesa, notebooks (es decir, portátiles), tabletas, decodificadores, teléfonos como los llamados teléfonos "inteligentes", tabletas, televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivo de transmisión continua de vídeo o similares. En algunos casos, el dispositivo fuente 12 y el dispositivo destino 14 pueden estar equipados para la comunicación inalámbrica. Por lo tanto, el dispositivo fuente 12 y el dispositivo destino 14 pueden ser dispositivos de comunicación inalámbrica. El dispositivo fuente 12 es un ejemplo de dispositivo de codificación de vídeo (es decir, un dispositivo para codificar datos de vídeo). El dispositivo destino 14 es un ejemplo de dispositivo de decodificación de vídeo (por ejemplo, un dispositivo o aparato para decodificar datos de vídeo).

65 En el ejemplo de la FIG. 1, el dispositivo fuente 12 incluye una fuente de vídeo 18, medios de almacenamiento 20 configurados para almacenar datos de vídeo, un codificador de vídeo 22 y una interfaz de salida 24. El

dispositivo destino 14 incluye una interfaz de entrada 26, un medio de almacenamiento 28 configurado para almacenar datos de vídeo codificados, un decodificador de vídeo 30 y un dispositivo de visualización 32. En otros ejemplos, el dispositivo fuente 12 y el dispositivo destino 14 incluyen otros componentes o disposiciones. Por ejemplo, el dispositivo fuente 12 puede recibir datos de vídeo de una fuente de vídeo externa, como una cámara externa. Del mismo modo, el dispositivo destino 14 puede interactuar con un dispositivo de visualización externo, en lugar de incluir un dispositivo de visualización integrado.

El sistema ilustrado 10 de la FIG. 1 es sólo un ejemplo. Las técnicas de procesamiento de datos de vídeo pueden realizarse utilizando cualquier dispositivo o aparato de codificación y/o decodificación de vídeo digital. Aunque generalmente las técnicas de esta divulgación son realizadas por un dispositivo de codificación de vídeo y un dispositivo de decodificación de vídeo, las técnicas también pueden ser realizadas por un codificador/decodificador de vídeo combinado, típicamente conocido como "CODEC". El dispositivo fuente 12 y el dispositivo destino 14 son solo ejemplos de tales dispositivos de codificación en los que el dispositivo fuente 12 genera datos de vídeo codificados para su transmisión al dispositivo destino 14. En algunos ejemplos, el dispositivo fuente 12 y el dispositivo destino 14 funcionan de manera sustancialmente simétrica, de modo que cada uno de los dispositivos de origen 12 y dispositivo de destino 14 incluyen componentes de codificación y decodificación de vídeo. Por lo tanto, el sistema 10 puede soportar la transmisión de vídeo unidireccional o bidireccional entre el dispositivo fuente 12 y el dispositivo destino 14, por ejemplo, para la transmisión continua de vídeo, la reproducción de vídeo, la emisión de vídeo o la videotelefonía.

La fuente de vídeo 18 del dispositivo fuente 12 puede incluir un dispositivo de captura de vídeo, como una cámara de vídeo, un archivo de vídeo que contenga vídeo capturado previamente y/o una interfaz de alimentación de vídeo para recibir datos de vídeo de un proveedor de contenidos de vídeo. Como alternativa adicional, la fuente de vídeo 18 puede generar datos basados en gráficos por ordenador como vídeo de origen, o como una combinación de vídeo en directo, vídeo archivado y vídeo generado por ordenador. El dispositivo fuente 12 puede comprender uno o más medios de almacenamiento de datos (por ejemplo, medios de almacenamiento 20) configurados para almacenar los datos de vídeo. Las técnicas descritas en esta divulgación pueden ser aplicables a la codificación de vídeo en general, y pueden aplicarse a aplicaciones inalámbricas y/o cableadas. En cada caso, el vídeo capturado, precapturado o generado por ordenador puede ser codificado por el codificador de vídeo 22. La interfaz de salida 24 puede emitir la información de vídeo codificada en un medio legible por ordenador 16.

El dispositivo destino 14 puede recibir los datos de vídeo codificados para ser decodificados a través de un medio legible por ordenador 16. El medio legible por ordenador 16 puede comprender cualquier tipo de medio o dispositivo capaz de trasladar los datos de vídeo codificados desde el dispositivo fuente 12 hasta el dispositivo destino 14. En algunos ejemplos, el medio legible por ordenador 16 comprende un medio de comunicación que permite que el dispositivo fuente 12 transmita datos de vídeo codificados directamente al dispositivo destino 14 en tiempo real. Los datos de vídeo codificados pueden modularse de acuerdo con un estándar de comunicación, como un protocolo de comunicación inalámbrica, y transmitirse al dispositivo destino 14. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrico o cableado, como un espectro de frecuencia de radio (RF) o una o más líneas de transmisión físicas. El medio de comunicación puede formar parte de una red basada en paquetes, como una red de área local, una red de área amplia o una red global como Internet. El medio de comunicación puede incluir enrutadores, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo fuente 12 hasta el dispositivo destino 14. El dispositivo destino 14 puede comprender uno o más medios de almacenamiento de datos configurados para almacenar datos de vídeo codificados y datos de vídeo decodificados.

En algunos ejemplos, los datos codificados (por ejemplo, datos de vídeo codificados) pueden emitirse desde la interfaz de salida 24 a un dispositivo de almacenamiento. Del mismo modo, se puede acceder a los datos codificados desde el dispositivo de almacenamiento utilizando la interfaz de entrada 26. El dispositivo de almacenamiento puede incluir cualquiera de una variedad de medios de almacenamiento de datos distribuidos o de acceso local, como un disco duro, discos Blu-ray, DVD, CD-ROM, memoria flash, memoria volátil o no volátil, o cualesquier otros medios de almacenamiento digitales adecuados para almacenar datos de vídeo codificados. En un ejemplo adicional, el dispositivo de almacenamiento puede corresponder a un servidor de archivos u otro dispositivo de almacenamiento intermedio que puede almacenar el vídeo codificado generado por el dispositivo fuente 12. El dispositivo destino 14 puede acceder a los datos de vídeo almacenados desde el dispositivo de almacenamiento a través de transmisión continua o descarga. El servidor de archivos puede ser cualquier tipo de servidor capaz de almacenar datos de vídeo codificados y transmitir esos datos de vídeo codificados al dispositivo destino 14. Ejemplos de servidores de archivos incluyen un servidor web (por ejemplo, para un sitio web), un servidor FTP, dispositivos de almacenamiento conectado a la red (NAS) o una unidad de disco local. El dispositivo destino 14 puede acceder a los datos de vídeo codificados a través de cualquier conexión de datos estándar, incluida una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión Wi-Fi), una conexión por cable (por ejemplo, DSL, módem de cable, etc.) o una combinación de ambos que sea adecuada para acceder a datos de vídeo codificados almacenados en un servidor de archivos. La transmisión de datos de vídeo codificados desde el dispositivo de almacenamiento puede ser una transmisión de transmisión continua, una transmisión de descarga o una combinación de ambas.

Las técnicas de esta divulgación pueden aplicarse a la codificación de vídeo en soporte de cualquiera de una variedad de aplicaciones multimedia, como emisiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo en transmisión continua por Internet, como transmisión continua dinámica adaptativa a través de HTTP (DASH), vídeo digital que está codificado en un medio de almacenamiento de datos, decodificación de vídeo digital almacenado en un medio de almacenamiento de datos, u otras aplicaciones. En algunos ejemplos, el sistema 10 puede configurarse para soportar la transmisión de vídeo unidireccional o bidireccional para soportar aplicaciones como la transmisión continua de vídeo, la reproducción de vídeo, la emisión de vídeo y/o la telefonía de vídeo.

El medio legible por ordenador 16 puede incluir medios transitorios, como una transmisión inalámbrica de emisión o de red por cable, o medios de almacenamiento (es decir, medios de almacenamiento no transitorios), como un disco duro, una unidad flash, un disco compacto, un disco de vídeo digital, un disco Blu-ray u otros medios legibles por ordenador. En algunos ejemplos, un servidor de red (no mostrado) puede recibir datos de vídeo codificados del dispositivo fuente 12 y proporcionar los datos de vídeo codificados al dispositivo destino 14, por ejemplo, a través de la transmisión de red. Similarmente, un dispositivo informático de una instalación de producción de medios, como una instalación de estampado de discos, puede recibir datos de vídeo codificados del dispositivo fuente 12 y producir un disco que contenga los datos de vídeo codificados. Por lo tanto, puede entenderse que el medio legible por ordenador 16 incluye uno o más medios legibles por ordenador de diversas formas, en diversos ejemplos.

La interfaz de entrada 26 del dispositivo destino 14 recibe información del medio legible por ordenador 16. La información del medio legible por ordenador 16 puede incluir información de sintaxis definida por el codificador de vídeo 22 del codificador de vídeo 22, que también es utilizada por el decodificador de vídeo 30, que incluye elementos de sintaxis que describen las características y/o el procesamiento de bloques y otras unidades codificadas, por ejemplo, grupos de imágenes (GOPs). Los medios de almacenamiento 28 pueden almacenar datos de vídeo codificados recibidos por la interfaz de entrada 26. El dispositivo de visualización 32 muestra los datos de vídeo decodificados a un usuario. El dispositivo de visualización 32 puede comprender cualquiera de una variedad de dispositivos de visualización, como un tubo de rayos catódicos (CRT), 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 de vídeo 22 y el decodificador de vídeo 30 pueden implementarse cada uno como cualquiera de una variedad de circuitos de codificador o decodificador adecuados, como uno o más microprocesadores, procesadores de señales digitales (DSPs), circuitos integrados de aplicaciones específicas (ASICs), matrices de puertas programables en campo (FPGAs), lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. Cuando las técnicas se implementan parcialmente en el software, un dispositivo puede almacenar instrucciones para el software en un medio adecuado, no transitorio y legible por ordenador y puede ejecutar las instrucciones en hardware utilizando uno o más procesadores para realizar las técnicas de esta divulgación. Cada uno de los codificadores de vídeo 22 y decodificadores de vídeo 30 puede estar incluido en uno o más codificadores o decodificadores, cualquiera de los cuales puede estar integrado como parte de un codificador/decodificador combinado (CODEC) en un dispositivo respectivo.

En algunos ejemplos, el codificador de vídeo 22 y el decodificador de vídeo 30 pueden funcionar de acuerdo con un estándar de codificación de vídeo. Entre los ejemplos de estándares de codificación de vídeo se incluyen, pero no están limitados a, 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 de Codificación de Vídeo Escalable (SVC) y Codificación de Vídeo de Múltiples Vistas (MVC). El estándar de codificación de vídeo de Codificación de Vídeo de Alta Eficiencia (HEVC) o ITU-T H.265 incluyendo sus extensiones de codificación de rango y contenido de pantalla, codificación de vídeo 3D (3D-HEVC) y extensiones multivista (MV-HEVC) y extensión escalable (SHVC) ha sido desarrollado por el equipo de colaboración conjunta en codificación de vídeo (JCT-VC) del grupo de expertos en codificación de vídeo (VCEG) de ITU-T y el Grupo de Expertos en Cinematografía de ISO/IEC.

En HEVC y otras especificaciones de codificación de vídeo, una secuencia de vídeo normalmente incluye una serie de imágenes. Las imágenes también pueden denominarse "fotogramas". Una imagen puede incluir tres matrices de muestras, denotadas  $S_L$ ,  $S_{Cb}$  y  $S_{Cr}$ .  $S_L$  es una matriz bidimensional (es decir, un bloque) de muestras de luma.  $S_{Cb}$  es una matriz bidimensional de muestras de crominancia de Cb.  $S_{Cr}$  es una matriz bidimensional de muestras de crominancia de Cr. Las muestras de crominancia también pueden denominarse en este documento muestras de "croma". En otros casos, una imagen puede ser monocromática y puede incluir solo una matriz de muestras de luma.

Más aún, en HEVC y otras especificaciones de codificación de vídeo, para generar una representación codificada de una imagen, el codificador de vídeo 22 puede generar un conjunto de unidades de árbol de codificación (CTUs). Cada una de las CTUs puede comprender un bloque de árbol de codificación de muestras de luma, dos bloques de árbol de codificación correspondientes de muestras de croma y estructuras de sintaxis

utilizadas para codificar las muestras de los bloques de árbol de codificación. En imágenes monocromáticas o imágenes que tienen tres planos de color separados, una CTU puede comprender un único bloque de árbol de codificación y estructuras de sintaxis utilizadas para codificar las muestras del bloque de árbol de codificación. Un bloque de árbol de codificación puede ser un bloque NxN de muestras. Una CTU también puede denominarse "bloque de árbol" o "unidad de codificación más grande" (LCU). Las CTU de HEVC pueden ser ampliamente análogas a los macrobloques de otros estándares, como H.264/AVC. Sin embargo, una CTU no está necesariamente limitada a un tamaño determinado y puede incluir una o más unidades de codificación (CUs). Un segmento puede incluir un número entero de CTUs ordenadas consecutivamente en un orden de escaneo ráster.

Si funciona de acuerdo con HEVC, para generar una CTU codificada, el codificador de vídeo 22 puede realizar recursivamente la partición de árbol cuaternario en los bloques de árbol de codificación de una CTU para dividir los bloques de árbol de codificación en bloques de codificación, de ahí el nombre de "unidades de árbol de codificación". Un bloque de codificación es un bloque NxN de muestras. Una CU puede comprender un bloque de codificación de muestras de luma y dos bloques de codificación correspondientes de muestras de croma de una imagen que tiene una matriz de muestras de luma, una matriz de muestras Cb y una matriz de muestras Cr, y estructuras de sintaxis utilizadas para codificar las muestras de los bloques de codificación. En imágenes monocromáticas o imágenes con tres planos de color separados, una CU puede comprender un único bloque de codificación y estructuras de sintaxis utilizadas para codificar las muestras del bloque de codificación.

Los datos de sintaxis dentro de un flujo de bits también pueden definir un tamaño para la CTU. Un segmento incluye un número de CTU consecutivas en orden de codificación. Un fotograma de vídeo o una imagen pueden ser particionados en uno o más segmentos. Como se mencionó anteriormente, cada bloque de árbol se puede dividir en unidades de codificación (CUs) de acuerdo con un árbol cuaternario. En general, una estructura de datos de árbol cuaternario incluye un nodo por CU, con un nodo raíz correspondiente al bloque de árbol. Si una CU se divide en cuatro subCU, el nodo correspondiente a la CU incluye cuatro nodos hoja, cada uno de los cuales corresponde a una de las subCU.

Cada nodo de la estructura de datos de árbol cuaternario puede proporcionar datos de sintaxis para la CU correspondiente. Por ejemplo, un nodo en el árbol cuaternario puede incluir un indicador de división, que indica si la CU correspondiente al nodo se divide en subCUs. Los elementos de sintaxis de una CU se pueden definir recursivamente y pueden depender de si la CU se divide en subCU. Si una CU no se divide más, se denomina CU hoja. Si un bloque de CU se divide aún más, generalmente se lo puede denominar CU no hoja. En algunos ejemplos de esta divulgación, cuatro subCU de una CU hoja pueden denominarse CU hoja incluso si no hay una división explícita de la CU hoja original. Por ejemplo, si una CU de tamaño 16x16 no se divide más, las cuatro subCU de 8x8 también se pueden denominar CU hoja, aunque la CU de 16x16 nunca se dividió.

Una CU tiene un propósito similar al de un macrobloque del estándar H.264, excepto que una CU no tiene una distinción de tamaño. Por ejemplo, un bloque de árbol puede dividirse en cuatro nodos hijo (también denominados subCUs), y cada nodo hijo puede ser a su vez un nodo padre y dividirse en otros cuatro nodos hijo. Un nodo hijo final no dividido, denominado nodo hoja del árbol cuaternario, comprende un nodo de codificación, también denominado CU hoja. Los datos de sintaxis asociados con un flujo de bits codificado pueden definir un número máximo de veces que se puede dividir un bloque de árbol, denominado profundidad máxima de CU, y también pueden definir un tamaño mínimo de los nodos de codificación. En consecuencia, un flujo de bits también puede definir una unidad de codificación más pequeña (SCU). Esta divulgación utiliza el término "bloque" para referirse a cualquiera de las CU, PU o TU, en el contexto de HEVC, o estructuras de datos similares en el contexto de otros estándares (por ejemplo, macrobloques y subbloques de los mismos en H.264/AVC).

Una CU incluye un nodo de codificación, así como unidades de predicción (PUs) y unidades de transformación (TUs) asociadas con el nodo de codificación. Un tamaño de la CU corresponde a un tamaño del nodo de codificación y puede ser, en algunos ejemplos, de forma cuadrada. En el ejemplo de HEVC, el tamaño de la CU puede oscilar entre 8x8 píxeles hasta el tamaño del bloque de árbol con un máximo de 64x64 píxeles o más. Cada CU puede contener una o más PU y una o más TU. Los datos de sintaxis asociados a una CU pueden describir, por ejemplo, la partición de la CU en una o varias PU. Los modos de partición pueden diferir entre si la CU está codificada en modo de omisión o directo, codificada en modo de intrapredicción o codificada en modo de interpredicción. Las PU se pueden particionar para que no tengan forma cuadrada. Los datos de sintaxis asociados con una CU también pueden describir, por ejemplo, la partición de la CU en una o más TU de acuerdo con un árbol cuaternario. Una TU puede tener forma cuadrada o no cuadrada (por ejemplo, rectangular).

El estándar HEVC permite transformaciones de acuerdo con TU. Las TU pueden ser diferentes para diferentes CU. Normalmente, el tamaño de las TU basadas en el tamaño de las PU dentro de una CU determinada definida para una LCU particionada, aunque no siempre es así. Las TU son normalmente del mismo tamaño o más pequeñas que las PU. En algunos ejemplos, las muestras residuales correspondientes a una CU pueden subdividirse en unidades más pequeñas utilizando una estructura de árbol cuaternario, a veces llamada "árbol

cuaternario residual" (RQT). Los nodos hoja del RQT pueden denominarse TU. Los valores de diferencia de píxeles asociados con las TU se pueden transformar para producir coeficientes de transformación, que se pueden cuantizar.

Una CU hoja puede incluir una o más PU. En general, una PU representa un área espacial correspondiente a la totalidad o a una porción de la CU correspondiente, y puede incluir datos para recuperar una muestra de referencia para la PU. Además, una PU incluye datos relacionados con la predicción. Por ejemplo, cuando la PU está codificada en intramodo, los datos para la PU pueden incluirse en un RQT, que puede incluir datos que describen un modo de intrapredicción para una TU correspondiente a la PU. Como otro ejemplo, cuando la PU está codificada en intermodo, la PU puede incluir datos que definen uno o más vectores de movimiento para la PU. Los datos que definen el vector de movimiento para una PU pueden describir, por ejemplo, un componente horizontal del vector de movimiento, un componente vertical del vector de movimiento, una resolución para el vector de movimiento (por ejemplo, una precisión de un cuarto de píxel o una precisión de un octavo de píxel), una imagen de referencia a la que apunta el vector de movimiento y/o una lista de imágenes de referencia (por ejemplo, Lista 0, Lista 1 o Lista C) para el vector de movimiento.

Una CU hoja que tiene una o más PU también puede incluir una o más TU. Las TU se pueden especificar utilizando un RQT (también conocido como estructura de árbol cuaternario de TU), como se discutió anteriormente. Por ejemplo, un indicador de división puede indicar si una CU hoja se divide en cuatro unidades de transformación. En algunos ejemplos, cada unidad de transformación puede dividirse a su vez en subTU adicionales. Cuando una TU no se divide más, puede denominarse TU hoja. Generalmente, para la intracodificación, todas las TU hoja que pertenecen a una CU hoja contienen datos residuales producidos a partir del mismo modo de intrapredicción. Es decir, generalmente se aplica el mismo modo de intrapredicción para calcular los valores predichos que se transformarán en todas las TU de una CU hoja. En el caso de la intracodificación, el codificador de vídeo 22 puede calcular un valor residual para cada TU hoja utilizando el modo de intrapredicción, como diferencia entre la porción de la CU correspondiente a la TU y el bloque original. Una TU no está necesariamente limitada al tamaño de una PU. Por lo tanto, las TU pueden ser más grandes o más pequeñas que una PU. Para la intracodificación, se puede colocar una PU con una TU hoja correspondiente para la misma CU. En algunos ejemplos, el tamaño máximo de una TU hoja puede corresponder al tamaño de la CU hoja correspondiente.

Por otra parte, las TU de las CU hoja también pueden estar asociadas con las respectivas estructuras RQT. Es decir, una CU hoja puede incluir un árbol cuaternario que indica cómo se particiona la CU hoja en las TU. El nodo raíz de un árbol cuaternario de TU generalmente corresponde a una CU hoja, mientras que el nodo raíz de un árbol cuaternario CU generalmente corresponde a un bloque de árbol (o LCU).

Como se ha comentado anteriormente, el codificador de vídeo 22 puede particionar un bloque de codificación de una CU en uno o más bloques de predicción. Un bloque de predicción es un bloque rectangular (es decir, cuadrado o no cuadrado) de muestras en las cuales se aplica la misma predicción. Una PU de una CU puede comprender un bloque de predicción de muestras de luma, dos bloques de predicción correspondientes de muestras de croma y estructuras de sintaxis utilizadas para predecir los bloques de predicción. En imágenes monocromáticas o imágenes con tres planos de color separados, una PU puede comprender un único bloque de predicción y estructuras de sintaxis utilizadas para predecir el bloque de predicción. El codificador de vídeo 22 puede generar bloques predictivos (por ejemplo, bloques predictivos de luma, Cb y Cr) para bloques de predicción (por ejemplo, bloques de predicción de luma, Cb y Cr) de cada PU de la CU.

El codificador de vídeo 22 puede utilizar la intrapredicción o la interpredicción para generar los bloques predictivos de una PU. Si el codificador de vídeo 22 utiliza la intrapredicción para generar los bloques predictivos de una PU, el codificador de vídeo 22 puede generar los bloques predictivos de la PU basado en muestras decodificadas de la imagen que incluyen la PU.

Después de que el codificador de vídeo 22 genera bloques predictivos (por ejemplo, bloques predictivos de luma, Cb y Cr) para una o más PU de una CU, el codificador de vídeo 22 puede generar uno o más bloques residuales para la CU. Por ejemplo, el codificador de vídeo 22 puede generar un bloque residual de luma para la CU. Cada muestra en el bloque residual de luma de la CU indica una diferencia entre una muestra de luma en uno de los bloques de luma predictivos de la CU y una muestra correspondiente en el bloque de codificación de luma original de la CU. Además, el codificador de vídeo 22 puede generar un bloque residual de Cb para la CU. Cada muestra en el bloque residual de Cb de una CU puede indicar una diferencia entre una muestra de Cb en uno de los bloques predictivos de Cb de la CU y una muestra correspondiente en el bloque de codificación de Cb original de la CU. El codificador de vídeo 22 también puede generar un bloque residual Cr para la CU. Cada muestra en el bloque residual de Cr de la CU puede indicar una diferencia entre una muestra de Cr en uno de los bloques predictivos de Cr de la CU y una muestra correspondiente en el bloque de codificación de Cr original de la CU.

Más aún, como se ha comentado anteriormente, el codificador de vídeo 22 puede utilizar la partición de árbol cuaternario para descomponer los bloques residuales (por ejemplo, los bloques residuales de luma, Cb y Cr)

de una CU en uno o más bloques de transformación (por ejemplo, bloques de transformación de luma, Cb y Cr). Un bloque de transformación es un bloque rectangular (por ejemplo, cuadrado o no cuadrado) de muestras en el que se aplica la misma transformación. Una unidad de transformación (TU) de una CU puede comprender un bloque de transformación de muestras de luma, dos bloques de transformación correspondientes de muestras de croma y estructuras de sintaxis utilizadas para transformar las muestras de bloques de transformación. Por lo tanto, cada TU de una CU puede tener un bloque de transformación de luma, un bloque de transformación Cb y un bloque de transformación Cr. El bloque de transformación de luma de la TU puede ser un subbloque del bloque residual de luma de la CU. El bloque de transformación Cb puede ser un subbloque del bloque residual Cb de la CU. El bloque de transformación Cr puede ser un subbloque del bloque residual Cr de la CU. En imágenes monocromáticas o imágenes con tres planos de color separados, una TU puede comprender un único bloque de transformación y estructuras de sintaxis utilizadas para transformar las muestras del bloque de transformación.

El codificador de vídeo 22 puede aplicar una o más transformaciones de un bloque de transformación de una TU para generar un bloque de coeficientes para la TU. Por ejemplo, el codificador de vídeo 22 puede aplicar una o más transformaciones a un bloque de transformación de luma de una TU para generar un bloque de coeficientes de luma para la TU. Un bloque de coeficientes puede ser una matriz bidimensional de coeficientes de transformación. Un coeficiente de transformación puede ser una cantidad escalar. El codificador de vídeo 22 puede aplicar una o más transformaciones a un bloque de transformación Cb de una TU para generar un bloque de coeficientes Cb para la TU. El codificador de vídeo 22 puede aplicar una o más transformaciones a un bloque de transformación Cr de una TU para generar un bloque de coeficientes Cr para la TU.

En algunos ejemplos, el codificador de vídeo 22 omite la aplicación de las transformaciones al bloque de transformación. En tales ejemplos, el codificador de vídeo 22 puede tratar los valores de muestra residuales de la misma manera que los coeficientes de transformación. Por lo tanto, en ejemplos en los que el codificador de vídeo 22 omite la aplicación de las transformaciones, la siguiente discusión sobre los coeficientes de transformación y los bloques de coeficientes puede ser aplicable a los bloques de transformación de muestras residuales.

Después de generar un bloque de coeficientes (por ejemplo, un bloque de coeficientes de luma, un bloque de coeficientes Cb o un bloque de coeficientes Cr), el codificador de vídeo 22 puede cuantizar el bloque de coeficientes para posiblemente reducir la cantidad de datos utilizados para representar el bloque de coeficiente, lo que podría proporcionar una compresión adicional. La cuantización generalmente se refiere a un proceso en el que un rango de valores se comprime a un único valor. Por ejemplo, la cuantización se puede realizar dividiendo un valor por una constante y luego redondeando al entero más cercano. Para cuantizar el bloque de coeficientes, el codificador de vídeo 22 puede cuantizar los coeficientes de transformación del bloque de coeficientes. Después de que el codificador de vídeo 22 cuantice un bloque de coeficientes, el codificador de vídeo 22 puede codificar la entropía de los elementos de sintaxis que indican los coeficientes de transformación cuantizados. Por ejemplo, el codificador de vídeo 22 puede realizar la Codificación Aritmética Binaria Adaptable al Contexto (CABAC) u otras técnicas de codificación de entropía en los elementos de sintaxis que indican los coeficientes de transformación cuantizados.

El codificador de vídeo 22 puede emitir un flujo de bits que incluya una secuencia de bits que forme una representación de las imágenes codificadas y los datos asociados. Por lo tanto, el flujo de bits comprende una representación codificada de datos de vídeo. El flujo de bits puede comprender una secuencia de unidades de capa de abstracción de red (NAL). Una unidad NAL es una estructura de sintaxis que contiene una indicación del tipo de datos en la unidad NAL y bytes que contienen esos datos en forma de una carga útil de secuencia de bytes sin procesar (RBSP) intercalada según sea necesario con bits de prevención de emulación. Cada una de las unidades NAL puede incluir un encabezado de unidad NAL y puede encapsular una RBSP. El encabezado de la unidad NAL puede incluir un elemento de sintaxis que indica un código de tipo de unidad NAL. El código de tipo de unidad NAL especificado por el encabezado de la unidad NAL de una unidad NAL indica el tipo de la unidad NAL. Una RBSP puede ser una estructura de sintaxis que contiene un número entero de bytes que se encapsula dentro de una unidad NAL. En algunos casos, una RBSP incluye cero bits.

El decodificador de vídeo 30 puede recibir un flujo de bits generado por el codificador de vídeo 22. El decodificador de vídeo 30 puede decodificar el flujo de bits para reconstruir imágenes de los datos de vídeo. Como parte de la decodificación del flujo de bits, el decodificador de vídeo 30 puede analizar el flujo de bits para obtener elementos de sintaxis del flujo de bits. El decodificador de vídeo 30 puede reconstruir las imágenes de los datos de vídeo basado, al menos en parte, en los elementos de sintaxis obtenidos del flujo de bits. El proceso de reconstrucción de los datos de vídeo puede ser generalmente recíproco al proceso realizado por el codificador de vídeo 22. Por ejemplo, el decodificador de vídeo 30 puede utilizar vectores de movimiento de las PU para determinar bloques predictivos para las PU de una CU actual. Además, el decodificador de vídeo 30 puede cuantizar inversamente los bloques de coeficientes de las TU de la CU actual. El decodificador de vídeo 30 puede realizar transformaciones inversas en los bloques de coeficientes para reconstruir los bloques de transformación de las TU de la CU actual. El decodificador de vídeo 30 puede reconstruir los bloques de codificación de la CU actual agregando las muestras de los bloques predictivos para las PU de la CU actual a



las muestras correspondientes de los bloques de transformación de las TU de la CU actual. Al reconstruir los bloques de codificación para cada CU de una imagen, el decodificador de vídeo 30 puede reconstruir la imagen.

A continuación se describen los conceptos comunes y ciertos aspectos de diseño de HEVC, centrándose en las técnicas para la partición de bloques. En HEVC, la unidad de codificación más grande en un segmento se denomina CTB. Un CTB se divide de acuerdo con una estructura de árbol cuaternario, cuyos nodos son unidades de codificación. La pluralidad de nodos en una estructura de árbol cuaternario incluye nodos hoja y nodos no hoja. Los nodos hoja no tienen nodos hijo en la estructura de árbol (es decir, los nodos hoja no se dividen más). Los nodos no hoja incluyen un nodo raíz de la estructura de árbol. El nodo raíz corresponde a un bloque de vídeo inicial de los datos de vídeo (por ejemplo, un CTB). Para cada nodo no raíz respectivo de la pluralidad de nodos, el nodo no raíz respectivo corresponde a un bloque de vídeo que es un subbloque de un bloque de vídeo correspondiente a un nodo padre en la estructura de árbol del nodo no raíz respectivo. Cada nodo no hoja respectivo de la pluralidad de nodos no hoja tiene uno o más nodos hijo en la estructura de árbol.

El tamaño de un CTB oscila desde 16x16 hasta 64x64 en el perfil principal HEVC (aunque técnicamente se pueden soportar tamaños de CTB de 8x8). Un CTB puede dividirse recursivamente en CU de forma de árbol cuaternario, como se describe en W. J. Han et al, "Improved vídeo Compression Efficiency through Flexible Unit Representation and Corresponding Extension of Coding Tools", IEEE Transaction on Circuits and Systems for vídeo Technology, vol. 20, no. 12, pp. 1709-1720, diciembre de 2010, y se muestra en la FIG. 2. Como se muestra en la FIG. 2, cada nivel de partición es un árbol cuaternario dividido en cuatro subbloques. El bloque negro es un ejemplo de un nodo hoja (es decir, un bloque que no se divide más).

En algunos ejemplos, una CU puede ser del mismo tamaño que un CTB, aunque una CU puede ser tan pequeña como 8x8. Cada CU se codifica con un modo de codificación, que podría ser, por ejemplo, un modo de intracodificación o un modo de intercodificación. También son posibles otros modos de codificación, incluidos los modos de codificación para el contenido de la pantalla (por ejemplo, modos de copia intrabloque, modos de codificación basados en paletas, etc.). Cuando una CU está intercodificada (es decir, se aplica el intermodo), la CU se puede particionar aún más en unidades de predicción (PU). Por ejemplo, una CU se puede particionar en 2 o 4 PU. En otro ejemplo, toda la CU se trata como una única PU cuando no se aplica una partición adicional. En ejemplos de HEVC, cuando dos PU están presentes en una CU, pueden ser rectángulos de la mitad de tamaño o dos rectángulos con un tamaño de  $\frac{1}{4}$  o  $\frac{3}{4}$  del tamaño de la CU.

En HEVC, hay ocho modos de partición para una CU codificada con modo de interpredicción, es decir, PART\_2Nx2N, PART\_2NxN, PART\_Nx2N, PART\_NxN, PART\_2NxN, PART\_2NxN, PART\_nLx2N y PART\_nRx2N, como se muestra en la FIG. 3. Como se muestra en la FIG. 3, una CU codificada con modo de partición PART\_2Nx2N no se divide más. Es decir, toda la CU se trata como una única PU (PU0). Una CU codificada con modo de partición PART\_2NxN se divide horizontalmente simétricamente en dos PU (PU0 y PU1). Una CU codificada con modo de partición PART\_Nx2N se divide verticalmente simétricamente en dos PU. Una CU codificada con modo de partición PART\_NxN se divide simétricamente en cuatro PU de igual tamaño (PU0, PU1, PU2, PU3).

Una CU codificada con modo de partición PART\_2NxN se divide horizontalmente asimétricamente en una PU0 (la PU superior) que tiene  $\frac{1}{4}$  del tamaño de la CU y una PU1 (la PU inferior) que tiene  $\frac{3}{4}$  del tamaño de la CU. Una CU codificada con el modo de partición PART\_2NxN se divide horizontalmente asimétricamente en una PU0 (la PU superior) que tiene  $\frac{3}{4}$  del tamaño de la CU y una PU1 (la PU inferior) que tiene  $\frac{1}{4}$  del tamaño de la CU. Una CU codificada con modo de partición PART\_nLx2N se divide verticalmente asimétricamente en una PU0 (la PU izquierda) que tiene  $\frac{1}{4}$  del tamaño de la CU y una PU1 (la PU derecha) que tiene  $\frac{3}{4}$  del tamaño de la CU. Una CU codificada con modo de partición PART\_nRx2N se divide verticalmente asimétricamente en una PU0 (la PU izquierda) que tiene  $\frac{3}{4}$  del tamaño de la CU y una PU1 (la PU derecha) que tiene  $\frac{1}{4}$  del tamaño de la CU.

Cuando una CU está intercodificada, un conjunto de información de movimiento (por ejemplo, vector de movimiento, dirección de predicción e imagen de referencia) está presente para cada PU. Además, cada PU está codificada con un modo de interpredicción único para derivar el conjunto de información de movimiento. Sin embargo, debe entenderse que incluso si dos PU están codificadas de manera única, aún pueden tener la misma información de movimiento en algunas circunstancias.

En J. An et al., "Block partitioning structure for next generation vídeo coding", International Telecommunication Union, COM16-C966, Sep. 2015 (en adelante, "VCEG proposal COM16-C966"), se propusieron técnicas de partición de árbol cuaternario árbol binario (QTBT) para futuros estándares de codificación de vídeo más allá de HEVC. Las simulaciones han demostrado que la estructura QTBT propuesta es más eficiente que la estructura de árbol cuaternario utilizado en HEVC.

En la estructura QTBT propuesta de la propuesta VCEG COM16-C966, un CTB se particiona primero utilizando técnicas de porcionamiento de árbol cuaternario, donde la división de árbol cuaternario de un nodo se puede iterar hasta que el nodo alcance el tamaño mínimo permitido de nodo hoja de árbol cuaternario. El tamaño

mínimo permitido de nodo hoja de árbol cuaternario puede indicarse al decodificador de vídeo utilizando el valor del elemento de sintaxis MinQTSIZE. Si el tamaño del nodo hoja de árbol cuaternario no es mayor que el tamaño máximo permitido del nodo raíz de árbol binario (por ejemplo, como se indica utilizando un elemento de sintaxis MaxBTSIZE), el nodo hoja de árbol cuaternario se puede particionar aún más utilizando la partición de árbol binario. La partición de árbol binario de un nodo se puede iterar hasta que el nodo alcance el tamaño mínimo permitido de nodo hoja de árbol binario (por ejemplo, como se indica utilizando un elemento de sintaxis MinBTSIZE) o la profundidad máxima permitida de árbol binario (por ejemplo, como se indica utilizando un elemento de sintaxis MaxBTDPTH). La propuesta VCEG COM16-C966 utiliza el término "CU" para referirse a los nodos hoja de árbol binario. En la propuesta VCEG COM16-C966, las CU se utilizan para la predicción (por ejemplo, intrapredicción, interpredicción, etc.) y transformaciones sin más particiones. En general, de acuerdo con las técnicas QTBT, hay dos tipos de división para la división de árboles binarios: división horizontal simétrica y división vertical simétrica. En cada caso, un bloque se divide dividiendo el bloque por la mitad, ya sea horizontal o verticalmente.

En un ejemplo de la estructura de partición QTBT, el tamaño de la CTU se establece en 128x128 (por ejemplo, un bloque de luma de 128x128 y dos bloques de croma correspondientes de 64x64), el MinQTSIZE se establece en 16x16, el MaxBTSIZE se establece en 64x64, el MinBTSIZE (tanto para el ancho como para el alto) se establece en 4 y el MaxBTDPTH se establece en 4. La partición de árbol cuaternario se aplica primero a la CTU para generar nodos hoja de árbol cuaternario. Los nodos de hoja de árbol cuaternario pueden tener un tamaño de 16x16 (es decir, el MinQTSIZE es 16x16) a 128x128 (es decir, el tamaño de CTU). De acuerdo con un ejemplo de partición QTBT, si el nodo de árbol cuaternario hoja es 128x128, el nodo de árbol cuaternario hoja no puede ser dividido aún más por el árbol binario, ya que el tamaño del nodo de árbol cuaternario hoja excede el MaxBTSIZE (es decir, 64x64). De lo contrario, el nodo de árbol cuaternario hoja se divide aún más por el árbol binario. Por lo tanto, el nodo hoja de árbol cuaternario es también el nodo raíz de árbol binario y tiene la profundidad de árbol binario como 0. La profundidad de árbol binario que alcanza MaxBTDPTH (por ejemplo, 4) implica que no hay más división. El nodo de árbol binario que tiene un ancho igual a MinBTSIZE (por ejemplo, 4) implica que no hay más división horizontal. Del mismo modo, el nodo de árbol binario que tiene una altura igual a MinBTSIZE no implica ninguna división vertical adicional. Los nodos hoja de árbol binario (CU) se procesan posteriormente (por ejemplo, utilizando la realización de un proceso de predicción y un proceso de transformación) sin ninguna partición adicional.

La FIG. 4A ilustra un ejemplo de un bloque 50 (por ejemplo, un CTB) particionado utilizando técnicas de particionamiento QTBT. Como se muestra en la FIG. 4A, utilizando técnicas de partición QTBT, cada uno de los bloques resultantes se divide simétricamente a través del centro de cada bloque. La FIG. 4B ilustra la estructura de árbol correspondiente a la partición de bloques de la FIG. 4B. Las líneas continuas de la FIG. 4B indican la división de árbol cuaternario y las líneas punteadas indican la división de árboles binarios. En un ejemplo, en cada nodo de división (es decir, no hoja) de árbol binario, se señala un elemento de sintaxis (por ejemplo, un indicador) para indicar el tipo de división realizada (por ejemplo, horizontal o vertical), donde 0 indica división horizontal y 1 indica división vertical. Para la división de árbol cuaternario, no es necesario indicar el tipo de división, ya que la división de árbol cuaternario siempre divide un bloque horizontal y verticalmente en 4 subbloques con el mismo tamaño.

Como se muestra en la FIG. 4B, en el nodo 70, el bloque 50 se divide en los cuatro bloques 51, 52, 53 y 54, que se muestran en la FIG. 4A, utilizando la partición QT. El bloque 54 no se divide más y, por lo tanto, es un nodo hoja. En el nodo 72, el bloque 51 se divide a su vez en dos bloques utilizando la partición BT. Como se muestra en la FIG. 4B, el nodo 72 está marcado con un 1, lo que indica una división vertical. Como tal, la división en el nodo 72 da como resultado el bloque 57 y el bloque que incluye los bloques 55 y 56. Los bloques 55 y 56 se crean utilizando una división vertical adicional en el nodo 74. En el nodo 76, el bloque 52 se divide adicionalmente en dos bloques 58 y 59 utilizando la partición BT. Como se muestra en la FIG. 4B, el nodo 76 está marcado con un 1, lo que indica una división horizontal.

En el nodo 78, el bloque 53 se divide en 4 bloques de igual tamaño utilizando la partición QT. Los bloques 63 y 66 se crean a partir de esta partición QT y no se dividen más. En el nodo 80, el bloque superior izquierdo se divide primero utilizando la división vertical de árbol binario, lo que da como resultado el bloque 60 y un bloque vertical derecho. A continuación, el bloque vertical derecho se divide utilizando la división horizontal de árbol binario en los bloques 61 y 62. El bloque inferior derecho creado a partir de la división de árbol cuaternario en el nodo 78, se divide en el nodo 84 utilizando la división horizontal de árbol binario en los bloques 64 y 65.

Aunque la estructura QTBT descrita anteriormente muestra un mejor rendimiento de codificación que la estructura de árbol cuaternario utilizada en HEVC, la estructura QTBT carece de flexibilidad. Por ejemplo, en la estructura QTBT descrita anteriormente, un nodo de árbol cuaternario se puede dividir aún más con un árbol binario, pero un nodo de árbol binario no se puede dividir aún más con un árbol cuaternario. En otro ejemplo, tanto el árbol cuaternario como el árbol binario solo pueden lograr una división uniforme (es decir, dividir el centro del bloque), lo que no es eficiente cuando un objeto está en el centro de un bloque que se va a dividir. Por lo tanto, el rendimiento de codificación de QTBT puede ser deficiente para los futuros estándares de codificación de vídeo.

Para abordar los problemas mencionados anteriormente, se proponen las siguientes técnicas. Las siguientes técnicas se pueden aplicar individualmente. En otros ejemplos, cualquier combinación de las técnicas que se describen a continuación se pueden aplicar juntas.

Para lograr una partición más flexible para una CTU, se propone una estructura CU basada en MTT para reemplazar las estructuras CU basadas en QT, BT y/o QTBT. La estructura de partición MTT de esta divulgación sigue siendo una estructura de árbol recursiva. Sin embargo, se utilizan múltiples estructuras de partición diferentes (por ejemplo, tres o más). Por ejemplo, de acuerdo con las técnicas MTT de esta divulgación, se pueden utilizar tres o más estructuras de partición diferentes a cada profundidad de una estructura de árbol. En este contexto, la profundidad de un nodo en una estructura de árbol puede referirse a la longitud de la ruta (por ejemplo, el número de divisiones) desde el nodo hasta la raíz de la estructura de árbol. Tal como se utiliza en esta divulgación, una estructura de partición generalmente puede referirse a cuántos bloques diferentes se puede dividir un bloque. Por ejemplo, una estructura de partición de árbol cuaternario puede dividir un bloque en cuatro bloques, una estructura de partición de árbol binario puede dividir un bloque en dos bloques y una estructura de partición de árbol terciario puede dividir un bloque en tres bloques. Una estructura de partición puede tener múltiples tipos de partición diferentes, como se explicará con más detalle a continuación. Un tipo de partición puede definir adicionalmente cómo se divide un bloque, incluida la partición simétrica o asimétrica, la partición uniforme o no uniforme y/o la partición horizontal o vertical.

En un ejemplo, de acuerdo con las técnicas de esta divulgación, el codificador de vídeo 22 puede configurarse para recibir una imagen de datos de vídeo y particionar la imagen de los datos de vídeo en una pluralidad de bloques utilizando tres o más estructuras de partición diferentes, y codificar la pluralidad de bloques de la imagen de los datos de vídeo. Similarmente, el decodificador de vídeo 30 puede configurarse para recibir un flujo de bits que incluya una secuencia de bits que forme una representación de una imagen codificada de datos de vídeo, determinar una partición de la imagen codificada de los datos de vídeo en una pluralidad de bloques utilizando tres o más estructuras de partición diferentes y reconstruir la pluralidad de bloques de la imagen codificada de los datos de vídeo. En un ejemplo, la partición del fotograma de los datos de vídeo comprende la partición del fotograma de los datos de vídeo en la pluralidad de bloques utilizando las tres o más estructuras de partición diferentes, en las que se pueden utilizar al menos tres de las tres o más estructuras de partición diferentes en cada profundidad de una estructura de árbol que representa cómo se particiona el fotograma de los datos de vídeo. En un ejemplo, las tres o más estructuras de partición diferentes incluyen una estructura de partición de árbol terciario, y el codificador de vídeo 22 y/o el decodificador de vídeo 30 pueden configurarse para particionar uno de la pluralidad de bloques de datos de vídeo utilizando un tipo de partición de árbol terciario de la estructura de partición de árbol terciario, en la que la estructura de partición de árbol terciario divide el uno de la pluralidad de bloques de en tres subbloques sin dividir el uno de la pluralidad de bloques a través del centro. En un ejemplo adicional de la divulgación, las tres o más estructuras de partición diferentes incluyen además una estructura de partición de árbol cuaternario y una estructura de partición de árbol binario.

Por lo tanto, en un ejemplo, el codificador de vídeo 22 puede generar una representación codificada de un bloque de vídeo inicial (por ejemplo, un bloque de árbol de codificación o CTU) de datos de vídeo. Como parte de la generación de la representación codificada del bloque de vídeo inicial, el codificador de vídeo 22 determina una estructura de árbol que comprende una pluralidad de nodos. Por ejemplo, el codificador de vídeo 22 puede particionar un bloque de árbol utilizando la estructura de partición MTT de esta divulgación.

La pluralidad de nodos en la estructura de partición MTT incluye una pluralidad de nodos hoja y una pluralidad de nodos no hoja. Los nodos hoja no tienen nodos hijo en la estructura de árbol. Los nodos no hoja incluyen un nodo raíz de la estructura de árbol. El nodo raíz corresponde al bloque de vídeo inicial. Para cada nodo no raíz respectivo de la pluralidad de nodos, el nodo no raíz respectivo corresponde a un bloque de vídeo (por ejemplo, un bloque de codificación) que es un subbloque de un bloque de vídeo correspondiente a un nodo padre en la estructura de árbol del nodo no raíz respectivo. Cada nodo no hoja respectivo de la pluralidad de nodos no hoja tiene uno o más nodos hijo en la estructura de árbol. En algunos ejemplos, un nodo no hoja en un límite de imagen puede tener solo un nodo hijo debido a una división forzada y uno de los nodos hijo corresponde a un bloque fuera del límite de imagen.

De acuerdo con las técnicas de esta divulgación, para cada respectivo nodo no hoja de la estructura de árbol en cada nivel de profundidad de la estructura de árbol, hay una pluralidad de patrones de división permitidos (por ejemplo, estructura de partición) para el respectivo nodo no hoja. Por ejemplo, puede haber tres o más estructuras de partición permitidas para cada profundidad de la estructura de árbol. El codificador de vídeo 22 puede configurarse para particionar un bloque de vídeo correspondiente al nodo no hoja respectivo en bloques de vídeo correspondientes a los nodos hijo del nodo no hoja respectivo de acuerdo con una de las estructuras de partición de la pluralidad permitida. Cada estructura de partición permitida respectiva de la pluralidad de estructuras de partición permitidas puede corresponder a una forma diferente de particionar el bloque de vídeo correspondiente al nodo no hoja respectivo en bloques de vídeo correspondientes a los nodos hijo del nodo no hoja respectivo. Más aún, en este ejemplo, el codificador de vídeo 22 puede incluir la representación codificada

del bloque de vídeo inicial en un flujo de bits que comprende una representación codificada de los datos de vídeo.

En un ejemplo similar, el decodificador de vídeo 30 puede determinar una estructura de árbol que comprende una pluralidad de nodos. Al igual que en el ejemplo anterior, la pluralidad de nodos incluye una pluralidad de nodos hoja y una pluralidad de nodos no hoja. Los nodos hoja no tienen nodos hijo en la estructura de árbol. Los nodos no hoja incluyen un nodo raíz de la estructura de árbol. El nodo raíz corresponde a un bloque de vídeo inicial de los datos de vídeo. Para cada nodo no raíz respectivo de la pluralidad de nodos, el nodo no raíz respectivo corresponde a un bloque de vídeo que es un subbloque de un bloque de vídeo correspondiente a un nodo padre en la estructura de árbol del nodo no raíz respectivo. Cada nodo no hoja respectivo de la pluralidad de nodos no hoja tiene uno o más nodos hijo en la estructura de árbol. Para cada nodo no hoja respectivo de la estructura de árbol en cada nivel de profundidad de la estructura de árbol, hay una pluralidad de patrones de división permitidos para el nodo no hoja respectivo y el bloque de vídeo correspondiente al nodo no hoja respectivo se divide en bloques de vídeo correspondientes a los nodos hijo del nodo no hoja respectivo de acuerdo con uno de la pluralidad de los patrones de división permitidos. Cada patrón de división permitido respectivo de la pluralidad de patrones de división permitidos corresponde a una forma diferente de particionar el bloque de vídeo correspondiente al nodo no hoja respectivo en bloques de vídeo correspondientes a los nodos hijo del nodo no hoja respectivo. Más aún, en este ejemplo, para cada (o al menos uno) nodo hoja respectivo de la estructura de árbol, el decodificador de vídeo 30 reconstruye el bloque de vídeo correspondiente al nodo hoja respectivo.

En algunos de estos ejemplos, para cada nodo no hoja respectivo de la estructura de árbol que no sea el nodo raíz, la pluralidad de patrones de división permitidos (por ejemplo, estructuras de partición) para el nodo no hoja respectivo es independiente de la estructura de partición de acuerdo con la cual un bloque de vídeo correspondiente a un nodo padre del respectivo nodo no hoja se particiona en bloques de vídeo correspondientes a nodos hijo del nodo padre del nodo no hoja respectivo.

En otros ejemplos de la divulgación, en cada profundidad de la estructura de árbol, el codificador de vídeo 22 puede configurarse para dividir aún más los subárboles utilizando un tipo de partición particular de entre una de las tres estructuras de partición adicionales. Por ejemplo, el codificador de vídeo 22 se puede configurar para determinar un tipo de partición particular de QT, BT, árbol terciario (TT) y otras estructuras de partición. En un ejemplo, la estructura de partición QT puede incluir tipos de partición cuadrada de árbol cuaternario y rectangulares de árbol cuaternario. El codificador de vídeo 22 puede particionar un bloque cuadrado utilizando la partición cuadrada de árbol cuaternario dividiendo el bloque, por el centro tanto horizontal como verticalmente, en cuatro bloques cuadrados de igual tamaño. Del mismo modo, el codificador de vídeo 22 puede dividir un bloque rectangular (por ejemplo, no cuadrado) utilizando una partición rectangular de árbol cuaternario dividiendo el bloque rectangular, por el centro tanto horizontal como verticalmente, en cuatro bloques rectangulares de igual tamaño.

La estructura de partición BT puede incluir tipos de partición horizontal simétrica de árbol binario, vertical simétrica de árbol binario, horizontal no simétrica de árbol binario y vertical no simétrica de árbol binario. Para el tipo de partición horizontal simétrica de árbol binario, el codificador de vídeo 22 puede configurarse para dividir un bloque, por el centro del bloque horizontalmente, en dos bloques simétricos del mismo tamaño. Para el tipo de partición vertical simétrica de árbol binario, el codificador de vídeo 22 puede configurarse para dividir un bloque, por el centro del bloque verticalmente, en dos bloques simétricos del mismo tamaño. Para el tipo de partición horizontal no simétrica de árbol binario, el codificador de vídeo 22 puede configurarse para dividir un bloque, horizontalmente, en dos bloques de diferente tamaño. Por ejemplo, un bloque puede tener  $\frac{1}{4}$  del tamaño del bloque padre y el otro bloque puede ser  $\frac{3}{4}$  del tamaño de los bloques padre, como en el tipo de partición PART\_2NxN<sub>U</sub> o PART\_2NxN<sub>D</sub> de la FIG. 3. Para el tipo de partición vertical no simétrica de árbol binario, el codificador de vídeo 22 puede configurarse para dividir un bloque, verticalmente, en dos bloques de diferente tamaño. Por ejemplo, un bloque puede ser  $\frac{1}{4}$  del tamaño del bloque padre y el otro bloque puede ser  $\frac{3}{4}$  del tamaño de los bloques padre, como en el tipo de partición PART\_nLx2N o PART\_nRx2N de la FIG. 3.

En otros ejemplos, un tipo de partición de árbol binario asimétrico puede dividir un bloque padre en fracciones de diferente tamaño. Por ejemplo, un subbloque puede ser  $\frac{2}{3}$  del bloque padre y el otro subbloque puede ser  $\frac{1}{3}$  del bloque padre. Por supuesto, este tipo de partición puede ser vertical u horizontal.

La estructura de partición TT difiere de la de las estructuras QT o BT, en que la estructura de partición TT no divide un bloque por el centro. La región central del bloque permanece junta en el mismo subbloque. A diferencia de QT, que produce cuatro bloques, o árbol binario, que produce dos bloques, la división de acuerdo con una estructura de partición TT produce tres bloques. Entre los tipos de partición de ejemplo de acuerdo con la estructura de partición TT se incluyen los tipos de partición simétricos (tanto horizontales como verticales), así como los tipos de partición asimétricos (tanto horizontales como verticales). Más aún, los tipos de partición simétrica de acuerdo con la estructura de partición TT pueden ser desigual/no uniforme o igual/uniforme. Los tipos de partición asimétrica de acuerdo con la estructura de partición TT de esta divulgación son desigual/no uniforme. En un ejemplo de la divulgación, una estructura de partición TT puede incluir los siguientes tipos de

partición: tipos de partición árbol terciario simétrico horizontal igual/uniforme, árbol terciario simétrico vertical igual/uniforme, árbol terciario simétrico horizontal desigual/no uniforme, árbol terciario simétrico vertical desigual/no uniforme, árbol terciario asimétrico horizontal desigual/no uniforme y árbol terciario asimétrico vertical desigual/no uniforme.

En general, un tipo de partición de árbol terciario simétrico desigual/no uniforme es un tipo de partición que es simétrica alrededor de una línea central del bloque, pero donde al menos uno de los bloques resultantes, tres bloques, no tiene el mismo tamaño que los otros dos. Un ejemplo preferido es donde los bloques laterales son  $\frac{1}{4}$  del tamaño del bloque, y el bloque central es  $\frac{1}{2}$  del tamaño del bloque. Un tipo de partición de árbol terciario simétrico igual/uniforme es un tipo de partición que es simétrico alrededor de una línea central del bloque, y los bloques resultantes son todos del mismo tamaño. Dicha partición es posible si la altura o el ancho del bloque, dependiendo de una división vertical u horizontal, es un múltiplo de 3. Un tipo de partición de árbol terciario asimétrico desigual/no uniforme es un tipo de partición que no es simétrico con respecto a una línea central del bloque y donde al menos uno de los bloques resultantes no tiene el mismo tamaño que los otros dos.

La FIG. 5A es un diagrama conceptual que ilustra ejemplos de tipos de particiones horizontales de árbol terciario. La FIG. 5B es un diagrama conceptual que ilustra ejemplos de tipos de particiones verticales de árbol terciario. Tanto en la FIG. 5A como en la FIG. 5B, h representa la altura del bloque en muestras de luma o croma y w representa el ancho del bloque en muestras de luma o croma. Obsérvese que las respectivas "líneas centrales" en cada una de las particiones de árbol terciario en las FIG. 5A y 5B no representan el límite del bloque (es decir, las particiones de árbol terciario no dividen un bloque a través de la línea central). Más bien, las líneas centrales se muestran para representar si un tipo de partición determinado es simétrico o asimétrico en relación con la línea central del bloque original. Las líneas centrales representadas también están a lo largo de la dirección de la división.

Como se muestra en la FIG. 5A, el bloque 71 está particionado con un tipo de partición simétrica horizontal igual/uniforme. El tipo de partición simétrica horizontal, igual/uniforme produce mitades superior e inferior simétricas en relación con la línea central del bloque 71. El tipo de partición simétrica horizontal igual/uniforme produce tres subbloques de igual tamaño, cada uno con una altura de  $h/3$  y una anchura de w. El tipo de partición simétrica horizontal igual/uniforme es posible cuando la altura del bloque 71 es divisible uniformemente por 3.

El bloque 73 está particionado con un tipo de partición simétrica horizontal desigual/no uniforme. El tipo de partición simétrica horizontal desigual/no uniforme produce mitades superior e inferior simétricas en relación con la línea central del bloque 73. El tipo de partición simétrica horizontal desigual/no uniforme produce dos bloques de igual tamaño (por ejemplo, los bloques superior e inferior con una altura de  $h/4$ ) y un bloque central de un tamaño diferente (por ejemplo, un bloque central con una altura de  $h/2$ ). En un ejemplo de la divulgación, de acuerdo con el tipo de partición simétrica horizontal desigual/no uniforme, el área del bloque central es igual a las áreas combinadas de los bloques superior e inferior. En algunos ejemplos, se puede preferir el tipo de partición simétrica horizontal desigual/no uniforme para bloques que tienen una altura que es una potencia de 2 (por ejemplo, 2, 4, 8, 16, 32, etc.).

El bloque 75 está particionado con un tipo de partición asimétrica horizontal desigual/no uniforme. El tipo de partición asimétrica horizontal desigual/no uniforme no produce una mitad superior e inferior simétrica en relación con la línea central del bloque 75 (es decir, las mitades superior e inferior son asimétricas). En el ejemplo de la FIG. 5A, el tipo de partición asimétrica horizontal desigual/no uniforme produce un bloque superior con una altura de  $h/4$ , un bloque central con una altura de  $3h/8$  y un bloque inferior con una altura de  $3h/8$ . Por supuesto, se pueden utilizar otras disposiciones asimétricas.

Como se muestra en la FIG. 5B, el bloque 77 está particionado con un tipo de partición simétrica vertical igual/uniforme. El tipo de partición simétrica vertical igual/uniforme produce mitades izquierda y derecha simétricas en relación con la línea central del bloque 77. El tipo de partición simétrica vertical igual/uniforme produce tres subbloques de igual tamaño, cada uno con un ancho de  $w/3$  y una altura de h. El tipo de partición simétrica vertical igual/uniforme es posible cuando el ancho del bloque 77 es divisible uniformemente por 3.

El bloque 79 está particionado con un tipo de partición simétrica vertical desigual/no uniforme. El tipo de partición simétrica vertical desigual/no uniforme produce mitades izquierda y derecha simétricas en relación con la línea central del bloque 79. El tipo de partición simétrica vertical desigual/no uniforme produce mitades izquierda y derecha simétricas en relación con la línea central de 79. El tipo de partición simétrica vertical desigual/no uniforme produce dos bloques de igual tamaño (por ejemplo, los bloques izquierdo y derecho con un ancho de  $w/4$ ) y un bloque central de un tamaño diferente (por ejemplo, un bloque central con un ancho de  $w/2$ ). En un ejemplo de la divulgación, de acuerdo con el tipo de partición simétrica vertical desigual/no uniforme, el área del bloque central es igual a las áreas combinadas de los bloques izquierdo y derecho. En algunos ejemplos, se puede preferir el tipo de partición simétrica vertical desigual/no uniforme para bloques que tienen una anchura que es una potencia de 2 (por ejemplo, 2, 4, 8, 16, 32, etc.).

El bloque 81 está particionado con un tipo de partición asimétrica vertical desigual/no uniforme. El tipo de partición asimétrica vertical desigual/no uniforme no produce una mitad izquierda y derecha simétricas en relación con la línea central del bloque 81 (es decir, las mitades izquierda y derecha son asimétricas). En el ejemplo de la FIG. 5B, el tipo de partición asimétrica vertical desigual/no uniforme produce un bloque izquierdo con un ancho de  $w/4$ , un bloque central con un ancho de  $3w/8$  y un bloque inferior con un ancho de  $3w/8$ . Por supuesto, se pueden utilizar otras disposiciones asimétricas.

En los ejemplos en los que un bloque (por ejemplo, en un nodo de subárbol) se divide en un tipo de partición de árbol terciario no simétrico, el codificador de vídeo 22 y/o el decodificador de vídeo 30 pueden aplicar una restricción tal que dos de las tres particiones tengan el mismo tamaño. Dicha restricción puede corresponder a una limitación que debe cumplir el codificador de vídeo 22 al codificar datos de vídeo. Más aún, en algunos ejemplos, el codificador de vídeo 22 y el decodificador de vídeo 30 pueden aplicar una restricción por la cual la suma del área de dos particiones es igual al área de la partición restante cuando se divide de acuerdo con un tipo de partición de árbol terciario no simétrica. Por ejemplo, el codificador de vídeo 22 puede generar o el decodificador de vídeo 30 puede recibir una representación codificada del bloque de vídeo inicial que cumple con una restricción que especifica que cuando un bloque de vídeo correspondiente a un nodo de la estructura de árbol se particiona de acuerdo con un patrón de árbol terciario no simétrico, el nodo tiene un primer nodo hijo, un segundo nodo hijo, y un tercer nodo hijo, el segundo nodo hijo correspondiente a un bloque de vídeo entre bloques de vídeo correspondientes al primer y tercer nodo hijo, los bloques de vídeo correspondientes al primer y tercer nodo hijo tienen el mismo tamaño, y una suma de los tamaños de los bloques de vídeo correspondientes al primer y tercer nodo hijo es igual a un tamaño del bloque de vídeo correspondiente al segundo nodo hijo.

En algunos ejemplos de la divulgación, el codificador de vídeo 22 puede configurarse para seleccionar entre todos los tipos de partición antes mencionados para cada una de las estructuras de partición QT, BT y TT. En otros ejemplos, el codificador de vídeo 22 se puede configurar para determinar sólo un tipo de partición entre un subconjunto de los tipos de partición antes mencionados. Por ejemplo, se puede utilizar un subconjunto de los tipos de partición mencionados anteriormente (u otros tipos de partición) para ciertos tamaños de bloque o para ciertas profundidades de una estructura de árbol cuaternario. El subconjunto de tipos de partición soportados puede señalizarse en el flujo de bits para su uso por el decodificador de vídeo 30 o puede predefinirse de forma que el codificador de vídeo 22 y el decodificador de vídeo 30 puedan determinar los subconjuntos sin ninguna señalización.

En otros ejemplos, el número de tipos de partición soportados puede ser fijo para todas las profundidades en todas las CTU. Es decir, el codificador de vídeo 22 y el decodificador de vídeo 30 pueden estar preconfigurados para utilizar el mismo número de tipos de partición para cualquier profundidad de una CTU. En otros ejemplos, el número de tipos de particiones soportados puede variar y puede depender de la profundidad, el tipo de segmento u otra información codificada anteriormente. En un ejemplo, en la profundidad 0 o la profundidad 1 de la estructura de árbol, solo se utiliza la estructura de partición QT. A profundidades mayores que 1, se puede utilizar cada una de las estructuras de partición QT, BT y TT.

En algunos ejemplos, el codificador de vídeo 22 y/o el decodificador de vídeo 30 pueden aplicar restricciones preconfiguradas en los tipos de partición soportados para evitar la duplicación de particiones para una determinada región de una imagen de vídeo o una región de una CTU. En un ejemplo, cuando un bloque se divide con un tipo de partición no simétrica, el codificador de vídeo 22 y/o el decodificador de vídeo 30 pueden configurarse para que no dividan más el subbloque más grande que se divide del bloque actual. Por ejemplo, cuando un bloque cuadrado se divide de acuerdo con un tipo de partición no simétrica (por ejemplo, tipo de partición PART\_2NxN en la FIG. 3), el subbloque más grande entre todos los subbloques (por ejemplo, PU1 del tipo de partición PART\_2NxN en la FIG. 3) es el nodo hoja anotado y no se puede dividir más. Sin embargo, el subbloque más pequeño (por ejemplo, PU0 de tipo de partición PART\_2NxN en la FIG. 3) se puede dividir aún más.

Como otro ejemplo en el que se pueden aplicar restricciones en los tipos de partición soportados para evitar la duplicación de particiones para una determinada región, cuando un bloque se divide con un tipo de partición no simétrico, el subbloque más grande que se divide del bloque actual no se puede dividir más en la misma dirección. Por ejemplo, cuando un bloque cuadrado se divide en un tipo de partición no simétrica (por ejemplo, tipo de partición PART\_2NxN en la FIG. 3), el codificador de vídeo 22 y/o el decodificador de vídeo 30 pueden configurarse para que no dividan el subbloque grande entre todos los subbloques (por ejemplo, PU1 del tipo de partición PART\_2NxN en la FIG. 3) en la dirección horizontal. Sin embargo, el codificador de vídeo 22 y/o el decodificador de vídeo 30, en este ejemplo, pueden dividir PU1 de nuevo en la dirección vertical.

Como otro ejemplo en el que se pueden aplicar restricciones a los tipos de partición soportados para evitar dificultades en la división posterior, el codificador de vídeo 22 y/o el decodificador de vídeo 30 pueden configurarse para no dividir un bloque, ya sea horizontal o verticalmente, cuando la anchura/altura de un bloque no es una potencia de 2 (por ejemplo, cuando la altura de la anchura no es 2, 4, 8, 16, etc.).

Los ejemplos anteriores describen cómo se puede configurar el codificador de vídeo 22 para realizar la partición MTT de acuerdo con las técnicas de esta divulgación. El decodificador de vídeo 30 también puede aplicar la misma partición MTT que la realizada por el codificador de vídeo 22. En algunos ejemplos, la forma en que el codificador de vídeo 22 particionó una imagen de datos de vídeo puede determinarse aplicando el mismo conjunto de reglas predefinidas en el decodificador de vídeo 30. Sin embargo, en muchas situaciones, el codificador de vídeo 22 puede determinar una estructura de partición y un tipo de partición particulares que se utilizarán basados en los criterios de distorsión de tasa para la imagen particular de los datos de vídeo que se codifican. Como tal, para que el decodificador de vídeo 30 determine la partición de una imagen particular, el codificador de vídeo 22 puede señalar elementos de sintaxis en el flujo de bits codificado que indican cómo se deben particionar la imagen y las CTU de la imagen. El decodificador de vídeo 30 puede analizar dichos elementos de sintaxis y particionar la imagen y las CTU en consecuencia.

En un ejemplo de la divulgación, el codificador de vídeo 22 puede configurarse para señalar un subconjunto particular de tipos de partición soportados como un elemento de sintaxis de alto nivel, en un conjunto de parámetros de secuencia (SPS), un conjunto de parámetros de imagen (PPS), un encabezado de segmento, un conjunto de parámetros adaptables (APS) o cualquier otro conjunto de parámetros de sintaxis de alto nivel. Por ejemplo, el número máximo de tipos de partición y los tipos soportados pueden estar predefinidos o señalados en el flujo de bits como un elemento de sintaxis de alto nivel, en el conjunto de parámetros de secuencia (SPS), el conjunto de parámetros de imagen (PPS) o cualquier otro conjunto de parámetros de sintaxis de alto nivel. El decodificador de vídeo 30 puede configurarse para recibir y analizar dicho elemento de sintaxis para determinar el subconjunto particular de tipos de partición que se utilizan y/o el número máximo de estructuras de partición (por ejemplo, QT, BT, TT, etc.) y tipos que se soportan.

En algunos ejemplos, en cada profundidad, el codificador de vídeo 22 puede configurarse para señalar un índice que indica el tipo de partición seleccionado utilizado a esa profundidad de la estructura de árbol. Más aún, en algunos ejemplos, el codificador de vídeo 22 puede señalar de forma adaptativa un índice de tipo de partición en cada CU, es decir, el índice puede ser diferente para diferentes CU. Por ejemplo, el codificador de vídeo 22 puede establecer el índice del tipo de partición basado en uno o más cálculos de distorsión de tasa. En un ejemplo, la señalización del tipo de partición (por ejemplo, el índice del tipo de partición) se puede omitir si se cumple cierta condición. Por ejemplo, el codificador de vídeo 22 puede omitir la señalización del tipo de partición cuando solo hay un tipo de partición soportada asociado a una profundidad específica. En este ejemplo, cuando se acerca al límite de una imagen, una región que se va a codificar puede ser más pequeña que una CTU. En consecuencia, en este ejemplo, se puede forzar la división de las CTU para que se ajusten al límite de la imagen. En un ejemplo, solo se utiliza un árbol binario simétrico para la división forzada y no se señala ningún tipo de partición. En algunos ejemplos, a una profundidad determinada, el tipo de partición se puede derivar basado en la información codificada previamente, como el tipo de segmento, la profundidad de la CTU y la posición de la CU.

En otro ejemplo de la divulgación, para cada CU (nodo hoja), el codificador de vídeo 22 puede configurarse además para señalar un elemento de sintaxis (por ejemplo, un indicador de transform\_split de un bit) para indicar si se debe realizar una transformación en el mismo tamaño de la CU o no (es decir, el indicador indica si la TU es del mismo tamaño que la CU o está dividida aún más). En caso de que el indicador transform\_split se señale como verdadero, el codificador de vídeo 22 puede configurarse para dividir aún más el residuo de la CU en múltiples subbloques y la transformación se realiza en cada subbloque. El decodificador de vídeo 30 puede realizar el proceso recíproco.

En un ejemplo, cuando el indicador transform\_split se señala como verdadero, se realiza lo siguiente. Si la CU corresponde a un bloque cuadrado (es decir, la CU es cuadrado), entonces el codificador de vídeo 22 divide el residuo en cuatro subbloques cuadrados utilizando la división de árbol cuaternario, y la transformación se realiza en cada subbloque cuadrado. Si la CU corresponde a un bloque no cuadrado, por ejemplo,  $M \times N$ , entonces el codificador de vídeo 22 divide el residuo en dos subbloques, y el tamaño del subbloque es  $0.5M \times N$  cuando  $M > N$ , y  $M \times 0.5N$  cuando  $M < N$ . Como otro ejemplo, cuando el indicador transform\_split se señala como verdadero y la CU corresponde a un bloque no cuadrado, por ejemplo,  $M \times N$ , (es decir, la CU no es cuadrada), el codificador de vídeo 22 puede configurarse para dividir el residuo en subbloques con tamaño  $K \times K$ , y la transformación cuadrada  $K \times K$  se utiliza para cada subbloque, donde  $K$  es igual al factor máximo de  $M$  y  $N$ . Como otro ejemplo, no se señala ningún indicador de transform\_split cuando una CU es un bloque cuadrado.

En algunos ejemplos, no se señala ningún indicador de división y solo se utiliza una transformación con un tamaño derivado para cuando hay residuos en la CU después de la predicción. Por ejemplo, en una CU con un tamaño igual a  $M \times N$ , se utiliza la transformada cuadrada  $K \times K$ , donde  $K$  es igual al factor máximo de  $M$  y  $N$ . Por lo tanto, en este ejemplo, para una CU con un tamaño de  $16 \times 8$ , la misma transformación de  $8 \times 8$  se puede aplicar a dos subbloques de  $8 \times 8$  de datos residuales de la CU. Un "indicador de división" es un elemento de sintaxis que indica que un nodo en una estructura de árbol tiene nodos hijo en la estructura de árbol.

En algunos ejemplos, para cada CU, si la CU no se divide en un árbol cuaternario cuadrado o en un árbol binario simétrico, el codificador de vídeo 22 está configurado para establecer siempre el tamaño de la transformación igual al tamaño de la partición (por ejemplo, el tamaño de la CU).

Los resultados de la simulación han demostrado que, en comparación con el software de referencia JEM-3.1, el rendimiento de la codificación utilizando las técnicas MTT de divulgación, en el caso de acceso aleatorio, ha mejorado. En promedio, las simulaciones han demostrado que las técnicas MTT de esta divulgación han proporcionado una reducción del 3.18 % en la tasa de bits y la distorsión (BD) con solo un aumento moderado del tiempo de codificación. Las simulaciones han demostrado que las técnicas MTT de esta divulgación proporcionan un buen rendimiento para resoluciones más altas, por ejemplo, una reducción de la tasa BD de luma del 4.20 % y del 4.89 % para las pruebas de Clase A1 y Clase A2. Las clases A1 y A2 son ejemplos de secuencias de prueba con resolución 4K.

Debe entenderse que, para cada uno de los ejemplos anteriores descritos con referencia al codificador de vídeo 22, el decodificador de vídeo 30 puede configurarse para realizar un proceso recíproco. Con respecto a los elementos de sintaxis de señalización, el decodificador de vídeo 30 puede configurarse para recibir y analizar dichos elementos de sintaxis y particiones y decodificar los datos de vídeo asociados en consecuencia.

En un ejemplo específico de la divulgación, el decodificador de vídeo puede configurarse para particionar bloques de vídeo de acuerdo con tres estructuras de partición diferentes (QT, BT y TT), con cinco tipos de partición diferentes permitidos en cada profundidad. Los tipos de partición incluyen la partición de árbol cuaternario (estructura de partición QT), la partición horizontal de árbol binario (estructura de partición BT), la partición vertical de árbol binario (estructura de partición BT), la partición horizontal de árbol terciario del lado central (estructura de partición TT) y la partición vertical de árbol terciario del lado central (estructura de partición TT), como se muestra en las FIG. 5A-5E.

Las definiciones de los cinco tipos de particionamiento son las siguientes. Tenga en cuenta que el cuadrado se considera un caso especial de rectangular.

- Partición de árbol cuaternario: un bloque se divide a su vez en cuatro bloques rectangulares del mismo tamaño. La FIG. 6A muestra un ejemplo de partición de árbol cuaternario.
- Partición vertical de árbol binario: un bloque se divide verticalmente en dos bloques rectangulares del mismo tamaño. La FIG. 6B es un ejemplo de partición vertical de árbol binario.
- Partición horizontal de árbol binario: un bloque se divide horizontalmente en dos bloques rectangulares del mismo tamaño. La FIG. 6C es un ejemplo de partición horizontal de árbol binario.
- Partición vertical de árbol terciario del lado central: un bloque se divide verticalmente en tres bloques rectangulares para que los dos bloques laterales compartan el mismo tamaño, mientras que el tamaño del bloque central es la suma de los dos bloques laterales. La FIG. 6D es un ejemplo de partición vertical de árbol terciario del lado central.
- Partición horizontal de árbol terciario del lado central: un bloque se divide horizontalmente en tres bloques rectangulares para que los dos bloques laterales compartan el mismo tamaño, mientras que el tamaño del bloque central es la suma de los dos bloques laterales. La FIG. 6E es un ejemplo de partición horizontal de árbol terciario del lado central.

Para un bloque asociado con una profundidad particular, el codificador de vídeo 22 determina qué tipo de partición (incluida la no división adicional) se utiliza y señala el tipo de partición determinado explícita o implícitamente (por ejemplo, el tipo de partición puede derivarse de reglas predeterminadas) al decodificador de vídeo 30. El codificador de vídeo 22 puede determinar el tipo de partición que se utilizará basado en la comprobación del coste de distorsión de tasa para el bloque utilizando diferentes tipos de partición. Para obtener el coste de la distorsión de tasa, el codificador de vídeo 22 puede necesitar comprobar los tipos de partición posibles para el bloque, recursivamente.

La FIG. 7 es un diagrama conceptual que ilustra un ejemplo de partición de unidades de árbol de codificación (CTU). En otras palabras, la FIG. 7 ilustra la partición de un CTB 91 correspondiente a una CTU. En el ejemplo de la FIG. 7,

- En la profundidad 0, el CTB 91 (es decir, todo el CTB) se divide en dos bloques con partición horizontal de árbol binario (como lo indica la línea 93 con guiones separados por únicos puntos).
- En la profundidad 1:



- El bloque superior se divide en tres bloques con partición vertical de árbol terciario del lado central (como lo indican las líneas 95 y 86 con pequeños guiones).

5      

- El bloque inferior se divide en cuatro bloques con partición de árbol cuaternario (como lo indican las líneas 88 y 90 con guiones separados por dos puntos).

- En la profundidad 2:

10      

- El bloque del lado izquierdo del bloque superior a la profundidad 1 se divide en tres bloques con partición horizontal de árbol terciario del lado central (como lo indican las líneas 92 y 94 con guiones largos separados por guiones cortos).

- No hay más división para los bloques central y derecho del bloque superior en la profundidad 1.

15      

- No hay más división para los cuatro bloques del bloque inferior en la profundidad 1

Como se puede ver en el ejemplo de la FIG. 7, se utilizan tres estructuras de partición diferentes (BT, QT y TT) con cuatro tipos de partición diferentes (partición horizontal de árbol binario, partición vertical de árbol terciario del lado central, partición de árbol cuaternario y partición horizontal de árbol terciario del lado central).

20      En otro ejemplo, se pueden aplicar restricciones adicionales para bloques a una cierta profundidad o con un cierto tamaño. Por ejemplo, si la altura/anchura de un bloque es inferior a 16 píxeles, el bloque no se puede dividir con un árbol vertical/horizontal del lado central para evitar un bloque con una altura/anchura inferior a 4 píxeles.

25      Se han descrito diversos ejemplos. Ejemplos particulares de esta divulgación pueden utilizarse por separado o en combinación entre sí.

30      La FIG. 8 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo 22 que puede implementar las técnicas de esta divulgación. La FIG. 8 se proporciona con fines explicativos y no debe considerarse limitante de las técnicas tal como se ejemplifican y describen ampliamente en esta divulgación. Las técnicas de esta divulgación pueden ser aplicables a diversos estándares o métodos de codificación.

35      En el ejemplo de la FIG. 8, el codificador de vídeo 22 incluye una unidad de procesamiento de predicción 100, una memoria de datos de vídeo 101, una unidad de generación residual 102, una unidad de procesamiento de transformación 104, una unidad de cuantización 106, una unidad de cuantización inversa 108, una unidad de procesamiento de transformación inversa 110, una unidad de reconstrucción 112, una unidad de filtro 114, un búfer de imagen decodificada 116, y una unidad de codificación de entropía 118. La unidad de procesamiento de predicción 100 incluye una unidad de procesamiento de interpredicción 120 y una unidad de procesamiento de intrapredicción 126. La unidad de procesamiento de interpredicción 120 puede incluir una unidad de estimación de movimiento y una unidad de compensación de movimiento (no se muestra).

45      La memoria de datos de vídeo 101 puede configurarse para almacenar datos de vídeo que van a ser codificados por los componentes del codificador de vídeo 22. Los datos de vídeo almacenados en la memoria de datos de vídeo 101 pueden obtenerse, por ejemplo, de la fuente de vídeo 18. El búfer de imagen decodificada 116 puede ser una memoria de imagen de referencia que almacena datos de vídeo de referencia para su uso en la codificación de datos de vídeo mediante el codificador de vídeo 22, por ejemplo, en modos de intra o intercodificación. La memoria de datos de vídeo 101 y el búfer de imagen decodificada 116 pueden estar formados por cualquiera de una variedad de dispositivos de memoria, como la memoria dinámica de acceso aleatorio (DRAM), incluida la DRAM síncrona (SDRAM), la RAM magnetorresistiva (MRAM), la RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. La memoria de datos de vídeo 101 y el búfer de imagen decodificada 116 pueden ser proporcionados por el mismo dispositivo de memoria o por dispositivos de memoria independientes. En diversos ejemplos, la memoria de datos de vídeo 101 puede estar en el chip con otros componentes del codificador de vídeo 22, o fuera del chip en relación con esos componentes. La memoria de datos de vídeo 101 puede ser la misma que o parte de los medios de almacenamiento 20 de la FIG. 1.

60      El codificador de vídeo 22 recibe datos de vídeo. El codificador de vídeo 22 puede codificar cada CTU en un segmento de una imagen de los datos de vídeo. Cada una de las CTU puede estar asociada con bloques de árbol de codificación de luma (CTB) de igual tamaño y los CTBs correspondientes de la imagen. Como parte de la codificación de una CTU, la unidad de procesamiento de predicción 100 puede realizar particiones para dividir los CTB de la CTU en bloques progresivamente más pequeños. Los bloques más pequeños pueden ser bloques de codificación de la CU. Por ejemplo, la unidad de procesamiento de predicción 100 puede particionar un CTB asociado a una CTU de acuerdo con una estructura de árbol. De acuerdo con una o más técnicas de esta divulgación, para cada nodo no hoja respectivo de la estructura de árbol en cada nivel de profundidad de la estructura de árbol, hay una pluralidad de patrones de división permitidos para el nodo no hoja respectivo y

el bloque de vídeo correspondiente al nodo no hoja respectivo se particiona en bloques de vídeo correspondientes a los nodos hijo del nodo no hoja respectivo de acuerdo con una de la pluralidad de patrones de división permitidos. En un ejemplo, la unidad de procesamiento de predicción 100 u otra unidad de procesamiento del codificador de vídeo 22 puede configurarse para realizar cualquier combinación de las técnicas de partición MTT descritas anteriormente.

El codificador de vídeo 22 puede codificar las CU de una CTU para generar representaciones codificadas de las CU (es decir, las CU codificadas). Como parte de la codificación de una CU, la unidad de procesamiento de predicción 100 puede particionar los bloques de codificación asociados con la CU entre una o más PU de la CU. De acuerdo con las técnicas de esta divulgación, una CU solo puede incluir una única PU. Es decir, en algunos ejemplos de esta divulgación, una CU no se divide en bloques de predicción separados, sino que se realiza un proceso de predicción en toda la CU. Por lo tanto, cada CU puede estar asociada con un bloque de predicción de luma y los bloques de predicción de croma correspondientes. El codificador de vídeo 22 y el decodificador de vídeo 30 pueden soportar las CU que tienen diversos tamaños. Como se indicó anteriormente, el tamaño de una CU puede referirse al tamaño del bloque de codificación de luma de la CU y al tamaño de un bloque de predicción de luma. Como se ha comentado anteriormente, el codificador de vídeo 22 y el decodificador de vídeo 30 pueden soportar tamaños de CU definidos por cualquier combinación de los tipos de partición MTT de ejemplo descritos anteriormente.

La unidad de procesamiento de interpredicción 120 puede generar datos predictivos para una PU realizando interpredicción en cada PU de una CU. Como se explicó anteriormente, en algunos ejemplos de MTT de esta divulgación, una CU puede contener solo una única PU, es decir, la CU y la PU pueden ser sinónimos. Los datos predictivos para la PU pueden incluir bloques predictivos de la PU e información de movimiento para la PU. La unidad de procesamiento de interpredicción 120 puede realizar diferentes operaciones para una PU o una CU dependiendo de si la PU está en un segmento I, un segmento P o un segmento B. En un segmento I, todas las PU están intrapredichas. Por consiguiente, si la PU está en un segmento I, la unidad de procesamiento de interpredicción 120 no realiza la interpredicción en la PU. Por lo tanto, para los bloques codificados en modo I, el bloque predicho se forma utilizando la predicción espacial de bloques vecinos previamente codificados dentro de la misma imagen. Si una PU está en un segmento P, la unidad de procesamiento de interpredicción 120 puede utilizar la interpredicción unidireccional para generar un bloque predictivo de la PU. Si una PU está en un segmento B, la unidad de procesamiento de interpredicción 120 puede utilizar la interpredicción unidireccional o bidireccional para generar un bloque predictivo de la PU.

La unidad de procesamiento de intrapredicción 126 puede generar datos predictivos para una PU realizando una intrapredicción en la PU. Los datos predictivos de la PU pueden incluir bloques predictivos de la PU y diversos elementos de sintaxis. La unidad de procesamiento de intrapredicción 126 puede realizar intrapredicción en las PU en segmentos I, segmentos P y segmentos B.

Para realizar la intrapredicción en una PU, la unidad de procesamiento de intrapredicción 126 puede utilizar múltiples modos de intrapredicción para generar múltiples conjuntos de datos predictivos para la PU. La unidad de procesamiento de intrapredicción 126 puede utilizar muestras de bloques de muestra de las PU vecinas para generar un bloque predictivo para una PU. Las PU vecinas pueden estar por encima, por encima y a la derecha, por encima y a la izquierda, o a la izquierda de la PU, suponiendo un orden de codificación de izquierda a derecha y de arriba a abajo para las PU, las CU y las CTU. La unidad de procesamiento de intrapredicción 126 puede utilizar diversos números de modos de intrapredicción, por ejemplo, 33 modos de intrapredicción direccionales. En algunos ejemplos, el número de modos de intrapredicción puede depender del tamaño de la región asociada con la PU.

La unidad de procesamiento de predicción 100 puede seleccionar los datos predictivos para las PU de una CU entre los datos predictivos generados por la unidad de procesamiento de interpredicción 120 para las PU o los datos predictivos generados por la unidad de procesamiento de intrapredicción 126 para las PU. En algunos ejemplos, la unidad de procesamiento de predicción 100 selecciona los datos predictivos para las PU de la CU basado en las métricas de tasa/distorsión de los conjuntos de datos predictivos. Los bloques predictivos de los datos predictivos seleccionados pueden denominarse en este documento bloques predictivos seleccionados.

La unidad de generación residual 102 puede generar, basada en los bloques de codificación (por ejemplo, bloques de codificación de luma, Cb y Cr) para una CU y los bloques predictivos seleccionados (por ejemplo, bloques predictivos de luma, Cb y Cr) para las PU de la CU, bloques residuales (por ejemplo, bloques residuales de luma, Cb y Cr) para la CU. Por ejemplo, la unidad de generación residual 102 puede generar los bloques residuales de la CU de manera que cada muestra de los bloques residuales tenga un valor igual a una diferencia entre una muestra en un bloque de codificación de la CU y una muestra correspondiente en un bloque predictivo seleccionado correspondiente de una PU de la CU.

La unidad de procesamiento de transformación 104 puede realizar particiones de árbol cuaternario para particionar los bloques residuales asociados con una CU en bloques de transformación asociados con las TU de la CU. Por lo tanto, una TU puede estar asociada con un bloque de transformación de luma y dos bloques

de transformación de croma. Los tamaños y posiciones de los bloques de transformación de luma y croma de las TU de una CU pueden o no estar basados en los tamaños y posiciones de los bloques de predicción de las PU de la CU. Una estructura de árbol cuaternario conocida como "árbol cuaternario residual" (RQT) puede incluir nodos asociados con cada una de las regiones. Las TU de una CU pueden corresponder a nodos hoja del RQT. En otros ejemplos, la unidad de procesamiento de transformación 104 se puede configurar para particionar las TU de acuerdo con las técnicas MTT descritas anteriormente. Por ejemplo, el codificador de vídeo 22 puede no dividir aún más las CU en las TU utilizando una estructura RQT. Como tal, en un ejemplo, una CU incluye una única TU.

La unidad de procesamiento de transformación 104 puede generar bloques de coeficientes de transformación para cada TU de una CU aplicando una o más transformaciones a los bloques de transformación de la TU. La unidad de procesamiento de transformación 104 puede aplicar diversas transformaciones a un bloque de transformación asociado con una TU. Por ejemplo, la unidad de procesamiento de transformación 104 puede aplicar una transformada de coseno discreta (DCT), una transformada direccional o una transformada conceptualmente similar a un bloque de transformación. En algunos ejemplos, la unidad de procesamiento de transformación 104 no aplica transformaciones a un bloque de transformación. En estos ejemplos, el bloque de transformación se puede tratar como un bloque de coeficientes de transformación.

La unidad de cuantización 106 puede cuantizar los coeficientes de transformación en un bloque de coeficiente. El proceso de cuantización puede reducir la profundidad de bits asociada con algunos o todos los coeficientes de transformación. Por ejemplo, un coeficiente de transformación de  $n$  bits puede redondearse hacia abajo a un coeficiente de transformación de  $m$  bits durante la cuantización, donde  $n$  es mayor que  $m$ . La unidad de cuantización 106 puede cuantizar un bloque de coeficientes asociado a una TU de una CU basado en un valor de parámetro de cuantización (QP) asociado a la CU. El codificador de vídeo 22 puede ajustar el grado de cuantización aplicado a los bloques de coeficientes asociados a una CU ajustando el valor QP asociado con la CU. La cuantización puede introducir pérdida de información. Por lo tanto, los coeficientes de transformación cuantizados pueden tener menor precisión que los originales.

La unidad de cuantización inversa 108 y la unidad de procesamiento de transformación inversa 110 pueden aplicar la cuantización inversa y las transformaciones inversas a un bloque de coeficientes, respectivamente, para reconstruir un bloque residual a partir del bloque de coeficientes. La unidad de reconstrucción 112 puede agregar el bloque residual reconstruido a las muestras correspondientes de uno o más bloques predictivos generados por la unidad de procesamiento de predicción 100 para producir un bloque de transformación reconstruido asociado con una TU. Al reconstruir los bloques de transformación para cada TU de una CU de esta manera, el codificador de vídeo 22 puede reconstruir los bloques de codificación de la CU.

La unidad de filtro 114 puede realizar una o más operaciones de desbloqueo para reducir los artefactos de bloqueo en los bloques de codificación asociados con una CU. El búfer de imagen decodificada 116 puede almacenar los bloques de codificación reconstruidos después de que la unidad de filtro 114 realice la una o más operaciones de desbloqueo en los bloques de codificación reconstruidos. La unidad de procesamiento de interpredicción 120 puede utilizar una imagen de referencia que contenga los bloques de codificación reconstruidos para realizar interpredicción en las PU de otras imágenes. Además, la unidad de procesamiento de intrapredicción 126 puede utilizar bloques de codificación reconstruidos en el búfer de imagen decodificada 116 para realizar intrapredicción en otras PU en la misma imagen que la CU.

La unidad de codificación de entropía 118 puede recibir datos de otros componentes funcionales del codificador de vídeo 22. Por ejemplo, la unidad de codificación de entropía 118 puede recibir bloques de coeficientes de la unidad de cuantización 106 y puede recibir elementos de sintaxis de la unidad de procesamiento de predicción 100. La unidad de codificación de entropía 118 puede realizar una o más operaciones de codificación de entropía en los datos para generar datos codificados en entropía. Por ejemplo, la unidad de codificación de entropía 118 puede realizar una operación CABAC, una operación de codificación de longitud variable adaptable al contexto (CAVLC), una operación de codificación de longitud de variable a variable (V2V), una operación de codificación aritmética binaria adaptable al contexto (SBAC) basada en sintaxis, una operación de codificación de Entropía de Partición de Intervalo de Probabilidad (PIPE), una operación de codificación de Golomb Exponencial u otro tipo de operación de codificación de entropía en los datos. El codificador de vídeo 22 puede emitir un flujo de bits que incluya datos codificados en entropía generados por la unidad de codificación de entropía 118. Por ejemplo, el flujo de bits puede incluir datos que representan la estructura de partición de una CU de acuerdo con las técnicas de esta divulgación.

La FIG. 9 es un diagrama de bloques que ilustra un ejemplo de decodificador de vídeo 30 que está configurado para implementar las técnicas de esta divulgación. La FIG. 9 se proporciona con fines explicativos y no se limita a las técnicas ampliamente ejemplificadas y descritas en esta divulgación. A efectos explicativos, esta divulgación describe el decodificador de vídeo 30 en el contexto de la codificación HEVC. Sin embargo, las técnicas de esta divulgación pueden ser aplicables a otros estándares o métodos de codificación.

En el ejemplo de la FIG. 9, el decodificador de vídeo 30 incluye una unidad de decodificación de entropía 150, una memoria de datos de vídeo 151, una unidad de procesamiento de predicción 152, una unidad de cuantización inversa 154, una unidad de procesamiento de transformación inversa 156, una unidad de reconstrucción 158, una unidad de filtro 160 y un búfer de imagen decodificada 162. La unidad de procesamiento de predicción 152 incluye una unidad de compensación de movimiento 164 y una unidad de procesamiento de intrapredicción 166. En otros ejemplos, el decodificador de vídeo 30 puede incluir más, menos o diferentes componentes funcionales.

La memoria de datos de vídeo 151 puede almacenar datos de vídeo codificados, como un flujo de bits de vídeo codificado, para ser decodificados por los componentes del decodificador de vídeo 30. Los datos de vídeo almacenados en la memoria de datos de vídeo 151 pueden obtenerse, por ejemplo, de un medio legible por ordenador 16, por ejemplo, de una fuente de vídeo local, como una cámara, a través de la comunicación de datos de vídeo por cable o de red inalámbrica, o accediendo a medios físicos de almacenamiento de datos. La memoria de datos de vídeo 151 puede formar un búfer de imagen codificada (CPB) que almacena datos de vídeo codificados de un flujo de bits de vídeo codificado. El búfer de imagen decodificada 162 puede ser una memoria de imagen de referencia que almacena datos de vídeo de referencia para su uso en la decodificación de datos de vídeo utilizando el decodificador de vídeo 30, por ejemplo, en modos de intracodificación o intercodificación, o para la salida. La memoria de datos de vídeo 151 y el búfer de imagen decodificada 162 pueden estar formados por cualquiera de una variedad de dispositivos de memoria, como la memoria dinámica de acceso aleatorio (DRAM), incluida la DRAM síncrona (SDRAM), la RAM magnetorresistiva (MRAM), la RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. La memoria de datos de vídeo 151 y el búfer de imagen decodificada 162 pueden ser proporcionados por el mismo dispositivo de memoria o por dispositivos de memoria separados. En diversos ejemplos, la memoria de datos de vídeo 151 puede estar en el chip con otros componentes del decodificador de vídeo 30, o fuera del chip en relación con esos componentes. La memoria de datos de vídeo 151 puede ser la misma que los medios de almacenamiento 28 de la FIG. 1 o formar parte de él.

La memoria de datos de vídeo 151 recibe y almacena datos de vídeo codificados (por ejemplo, unidades NAL) de un flujo de bits. La unidad de decodificación de entropía 150 puede recibir datos de vídeo codificados (por ejemplo, unidades NAL) de la memoria de datos de vídeo 151 y puede analizar las unidades NAL para obtener elementos de sintaxis. La unidad de decodificación de entropía 150 puede decodificar elementos de sintaxis codificados en entropía en las unidades NAL. La unidad de procesamiento de predicción 152, la unidad de cuantización inversa 154, la unidad de procesamiento de transformación inversa 156, la unidad de reconstrucción 158 y la unidad de filtro 160 pueden generar datos de vídeo decodificados basados en los elementos de sintaxis extraídos del flujo de bits. La unidad de decodificación de entropía 150 puede realizar un proceso generalmente recíproco al de la unidad de codificación de entropía 118.

De acuerdo con algunos ejemplos de esta divulgación, la unidad de decodificación de entropía 150, u otra unidad de procesamiento del decodificador de vídeo 30, puede determinar una estructura de árbol como parte de la obtención de los elementos de sintaxis del flujo de bits. La estructura de árbol puede especificar cómo se divide un bloque de vídeo inicial, como un CTB, en bloques de vídeo más pequeños, como unidades de codificación. De acuerdo con una o más técnicas de esta divulgación, para cada nodo no hoja respectivo de la estructura de árbol en cada nivel de profundidad de la estructura de árbol, hay una pluralidad de tipos de partición permitidos para el nodo no hoja respectivo y el bloque de vídeo correspondiente al nodo no hoja respectivo se particiona en bloques de vídeo correspondientes a los nodos hijo del nodo no hoja respectivo de acuerdo con una de la pluralidad de patrones de división permitidos.

Además de obtener elementos de sintaxis del flujo de bits, el decodificador de vídeo 30 puede realizar una operación de reconstrucción en una CU no particionada. Para realizar la operación de reconstrucción en una CU, el decodificador de vídeo 30 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 de vídeo 30 puede reconstruir los bloques residuales de la CU. Como se mencionó anteriormente, en un ejemplo de divulgación, una CU incluye una única TU.

Como parte de la realización de una operación de reconstrucción en una TU de una CU, la unidad de cuantización inversa 154 puede cuantizar inversamente, es decir, descuantizar, bloques de coeficientes asociados con la TU. Después de que la unidad de cuantización inversa 154 cuantice inversamente un bloque de coeficientes, la unidad de procesamiento de transformación inversa 156 puede aplicar una o más transformaciones inversas al bloque de coeficientes para generar un bloque residual asociado con la TU. Por ejemplo, la unidad de procesamiento de transformación inversa 156 puede aplicar una DCT inversa, una transformada entera inversa, una transformada inversa de Karhunen-Loeve (KLT), una transformada rotacional inversa, una transformada direccional inversa u otra transformada inversa al bloque de coeficientes.

Si una CU o una PU se codifica utilizando la intrapredicción, la unidad de procesamiento de intrapredicción 166 puede realizar una intrapredicción para generar bloques predictivos de la PU. La unidad de procesamiento de intrapredicción 166 puede utilizar un modo de intrapredicción para generar los bloques predictivos de la PU.

basada en muestras de bloques espacialmente vecinos. La unidad de procesamiento de intrapredicción 166 puede determinar el modo de intrapredicción para la PU basado en uno o más elementos de sintaxis obtenidos del flujo de bits.

- 5 Si una PU se codifica utilizando interpredicción, la unidad de decodificación de entropía 150 puede determinar la información de movimiento para la PU. La unidad de compensación de movimiento 164 puede determinar, basado en la información de movimiento de la PU, uno o más bloques de referencia. La unidad de compensación de movimiento 164 puede generar, basado en el uno o más bloques de referencia, bloques predictivos (por ejemplo, luma predictiva, bloques Cb y Cr) para la PU. Como se ha comentado anteriormente, en un ejemplo de la divulgación utilizando la partición MTT, una CU puede incluir solo una única PU. Es decir, una CU no puede dividirse en múltiples PU.

- 15 La unidad de reconstrucción 158 puede utilizar bloques de transformación (por ejemplo, bloques de transformación de luma, Cb y Cr) para las TU de una CU y los bloques predictivos (por ejemplo, bloques de luma, Cb y Cr) de las PU de la CU, es decir, datos de intrapredicción o datos de interpredicción, según corresponda, para reconstruir los bloques de codificación (por ejemplo, bloques de codificación de luma, Cb y Cr) para la CU. Por ejemplo, la unidad de reconstrucción 158 puede agregar muestras de los bloques de transformación (por ejemplo, bloques de transformación de luma, Cb y Cr) a las muestras correspondientes de los bloques predictivos (por ejemplo, bloques predictivos de luma, Cb y Cr) para reconstruir los bloques de codificación (por ejemplo, bloques de codificación de luma, Cb y Cr) de la CU.

- 25 La unidad de filtro 160 puede realizar una operación de desbloqueo para reducir los artefactos de bloqueo asociados con los bloques de codificación de la CU. El decodificador de vídeo 30 puede almacenar los bloques de codificación de la CU en el búfer de imagen decodificada 162. El búfer de imagen decodificada 162 puede proporcionar imágenes de referencia para la compensación de movimiento posterior, la intrapredicción y la presentación en un dispositivo de visualización, como el dispositivo de visualización 32 de la FIG. 1. Por ejemplo, el decodificador de vídeo 30 puede realizar, basado en los bloques del búfer de imagen decodificada 162, operaciones de intrapredicción o de interpredicción para las PU de otras CU.

- 30 La FIG. 10A es un diagrama de flujo que ilustra un ejemplo de operación del codificador de vídeo 22, de acuerdo con una técnica de esta divulgación. En el ejemplo de la FIG. 10A, el codificador de vídeo 22 puede generar una representación codificada de un bloque de vídeo inicial (por ejemplo, un bloque de árbol de codificación) de datos de vídeo (200). Como parte de la generación de la representación codificada del bloque de vídeo inicial, el codificador de vídeo 22 determina una estructura de árbol que comprende una pluralidad de nodos. La pluralidad de nodos incluye una pluralidad de nodos hoja y una pluralidad de nodos no hoja. Los nodos hoja no tienen nodos hijo en la estructura de árbol. Los nodos no hoja incluyen un nodo raíz de la estructura de árbol. El nodo raíz corresponde al bloque de vídeo inicial. Para cada nodo no raíz respectivo de la pluralidad de nodos, el nodo no raíz respectivo corresponde a un bloque de vídeo (por ejemplo, un bloque de codificación) que es un subbloque de un bloque de vídeo correspondiente a un nodo padre en la estructura de árbol del nodo no raíz respectivo. Cada nodo no hoja respectivo de la pluralidad de nodos no hoja tiene uno o más nodos hijo en la estructura de árbol. Para cada nodo no hoja respectivo de la estructura de árbol en cada nivel de profundidad de la estructura de árbol, hay una pluralidad de tipos de partición permitidos de tres o más estructuras de partición (por ejemplo, estructuras de partición BT, QT y TT) para el nodo no hoja respectivo y el bloque de vídeo correspondiente al nodo no hoja respectivo se particiona en bloques de vídeo correspondientes a los nodos hijo del nodo no hoja respectivo de acuerdo con una de la pluralidad de Tipos de partición permitidos. Cada uno de los respectivos tipos de partición permitidos de la pluralidad de tipos de partición permitidos puede corresponder a una forma diferente de particionar el bloque de vídeo correspondiente al nodo no hoja respectivo en bloques de vídeo correspondientes a los nodos hijo del nodo no hoja respectivo. Más aún, en este ejemplo, el codificador de vídeo 22 puede incluir la representación codificada del bloque de vídeo inicial en un flujo de bits que comprende una representación codificada de los datos de vídeo (202).

- La FIG. 10B es un diagrama de flujo que ilustra un ejemplo de operación del decodificador de vídeo 30, de acuerdo con una técnica de esta divulgación. En el ejemplo de la FIG. 10B, el decodificador de vídeo 30 puede determinar una estructura de árbol que comprende una pluralidad de nodos (250). La pluralidad de nodos incluye una pluralidad de nodos hoja y una pluralidad de nodos no hoja: los nodos hoja no tienen nodos hijo en la estructura de árbol. Los nodos no hoja incluyen un nodo raíz de la estructura de árbol. El nodo raíz corresponde a un bloque de vídeo inicial de los datos de vídeo. Para cada nodo no raíz respectivo de la pluralidad de nodos, el nodo no raíz respectivo corresponde a un bloque de vídeo que es un subbloque de un bloque de vídeo correspondiente a un nodo padre en la estructura de árbol del nodo no raíz respectivo. Cada nodo no hoja respectivo de la pluralidad de nodos no hoja tiene uno o más nodos hijo en la estructura de árbol. Para cada nodo no hoja respectivo de la estructura de árbol en cada nivel de profundidad de la estructura de árbol, hay una pluralidad de tipos de partición permitidos de tres o más estructuras de partición (por ejemplo, estructuras de partición BT, QT y TT) para el nodo no hoja respectivo y el bloque de vídeo correspondiente al nodo no hoja respectivo se particiona en bloques de vídeo correspondientes a los nodos hijo del nodo no hoja respectivo de acuerdo con una de la pluralidad de Tipos de partición permitidos. Cada tipo de partición permitido

respectivo de la pluralidad de tipos de partición permitidos corresponde a una forma diferente de particionar el bloque de vídeo correspondiente al nodo no hoja respectivo en bloques de vídeo correspondientes a los nodos hijo del nodo no hoja respectivo. Más aún, en este ejemplo, para cada (o al menos uno) nodo hoja respectivo de la estructura de árbol, el decodificador de vídeo 30 reconstruye el bloque de vídeo correspondiente al nodo hoja respectivo (252).

En los ejemplos de la FIG. 10A y la FIG. 10B, para cada nodo no hoja respectivo de la estructura de árbol que no sea el nodo raíz, la pluralidad de tipos de partición permitidos para el nodo no hoja respectivo puede ser independiente de un patrón de división de acuerdo con el cual un bloque de vídeo correspondiente a un nodo padre del nodo no hoja respectivo se particiona en bloques de vídeo correspondientes a nodos hijo del nodo padre del respectivo nodo no hoja. Por ejemplo, a diferencia de la propuesta VCEG COM16C966, si el bloque de vídeo de un nodo en particular se divide de acuerdo con un patrón de división de árbol binario, el bloque de vídeo de un nodo hijo del nodo en particular puede dividirse de acuerdo con un patrón de división de árbol cuaternario.

Más aún, en los ejemplos de la FIG. 10A y la FIG. 10B, para cada nodo no hoja respectivo de la estructura de árbol, la pluralidad de patrones de división permitidos para el respectivo nodo no hoja puede incluir dos o más de: un patrón de división de árbol cuaternario cuadrado, un patrón de división de árbol cuaternario rectangular, un patrón de división de árbol binario simétrico, un patrón de división de árbol binario no simétrico, un patrón de división de árbol terciario simétrico o un patrón de división de árbol terciario no simétrico.

Más aún, como se ha indicado anteriormente, sólo se utiliza un subconjunto de los tipos de partición mencionados anteriormente. El subconjunto de tipos de partición soportados puede estar señalizado en el flujo de bits o predefinido. Por lo tanto, en algunos ejemplos, el decodificador de vídeo 30 puede obtener, a partir de un flujo de bits, elementos de sintaxis que indican una pluralidad de patrones de división soportados. Similarmente, el codificador de vídeo 22 puede señalar, en el flujo de bits, una pluralidad de patrones de división soportados. En estos ejemplos, para cada nodo no hoja respectivo de la estructura de árbol, la pluralidad de patrones de división soportados puede incluir la pluralidad de patrones de división permitidos para el nodo no hoja respectivo. En estos ejemplos, los elementos de sintaxis que indican la pluralidad de patrones de división soportados se pueden obtener de (y señalar en) un flujo de bits, como en un conjunto de parámetros de secuencia (SPS) o un conjunto de parámetros de imagen (PPS), o un encabezado de segmento.

Como se indicó anteriormente, en algunos ejemplos, cuando el subárbol se divide en un árbol terciario no simétrico, se aplica una restricción de que las dos de las tres particiones tengan el mismo tamaño. En consecuencia, en algunos ejemplos, el decodificador de vídeo 30 puede recibir una representación codificada del bloque de vídeo inicial que cumple con una restricción que especifica que cuando un bloque de vídeo correspondiente a un nodo de la estructura de árbol se particiona de acuerdo con un patrón de árbol terciario no simétrico, los bloques de vídeo correspondientes a dos nodos hijo del nodo tienen el mismo tamaño. Del mismo modo, el codificador de vídeo 22 puede generar la representación codificada del bloque de vídeo inicial para cumplir con una restricción que especifica que cuando un bloque de vídeo correspondiente a un nodo de la estructura de árbol se particiona de acuerdo con un patrón de árbol terciario no simétrico, los bloques de vídeo correspondientes a dos nodos hijo del nodo tienen el mismo tamaño.

Como se indicó anteriormente, en algunos ejemplos, el número de tipos de partición admitidos puede ser fijo para todas las profundidades en todas las CTU. Por ejemplo, puede haber el mismo número de patrones de división permitidos en la pluralidad de patrones de división permitidos para cada nodo no hoja de la estructura de árbol. Adicionalmente, como se indicó anteriormente, en otros ejemplos, el número de tipos de partición soportados puede depender de la profundidad, el tipo de segmento, el tipo de CTU u otra información codificada anteriormente. Por ejemplo, para al menos un nodo no hoja de la estructura de árbol, el número de patrones de división permitidos en la pluralidad de patrones de división permitidos para el nodo no hoja depende de al menos uno de: una profundidad del nodo no hoja en la estructura de árbol, el tamaño del bloque de vídeo correspondiente al nodo no hoja en la estructura de árbol, un tipo de segmento o información previamente codificada.

En algunos ejemplos, cuando un bloque se divide con un tipo de partición no simétrica (por ejemplo, los tipos de partición de árbol binario no simétricos que se muestran en la FIG. 3, incluida PART\_2NxN, PART\_2NxND, PART\_nLx2N, PART\_nRx2N), el subbloque más grande que se divide del bloque actual no se puede dividir más. Por ejemplo, una restricción sobre cómo se codifica el bloque de vídeo inicial puede requerir que, cuando un bloque de vídeo correspondiente a cualquier nodo no hoja de la estructura de árbol se divida en una pluralidad de subbloques de acuerdo con un patrón de división no simétrico, el subbloque más grande de la pluralidad de subbloques corresponde a un nodo hoja de la estructura de árbol.

En algunos ejemplos, cuando un bloque se divide con un tipo de partición no simétrica, el subbloque más grande que se divide del bloque actual no se puede dividir más en la misma dirección. Por ejemplo, una restricción sobre cómo se codifica el bloque de vídeo inicial puede requerir que, cuando un bloque de vídeo correspondiente a cualquier nodo no hoja de la estructura de árbol se divide en una primera dirección en una

pluralidad de subbloques de acuerdo con un patrón de división no simétrico, la estructura de árbol no puede contener nodos correspondientes a subbloques del subbloque más grande de la pluralidad de subbloques que se dividen del subbloque más grande de la pluralidad de subbloques en la primera dirección.

5 En algunos ejemplos, cuando la anchura/altura de un bloque no es una potencia de 2, no se permite ninguna otra división horizontal/vertical. Por ejemplo, una restricción sobre cómo se codifica el bloque de vídeo inicial puede requerir que un nodo de la estructura de árbol correspondiente a un bloque de vídeo cuya altura o anchura no sea una potencia de 2 sea un nodo hoja.

10 En algunos ejemplos, en cada profundidad, se señala un índice del tipo de partición seleccionado en el flujo de bits. Por lo tanto, en algunos ejemplos, el codificador de vídeo 22 puede incluir, en el flujo de bits, un índice de un patrón de división de acuerdo con el cual un bloque de vídeo correspondiente a un nodo no hoja de la estructura de árbol se divide en bloques de vídeo correspondientes a nodos hijo del nodo no hoja. De manera similar, en algunos ejemplos, el decodificador de vídeo 30 puede obtener, a partir de un flujo de bits, un índice de un patrón de división de acuerdo con el cual un bloque de vídeo correspondiente a un nodo no hoja de la estructura de árbol se divide en bloques de vídeo correspondientes a nodos hijo del nodo no hoja.

En algunos ejemplos, para cada CU (nodo hoja), se señala aún más un indicador de transform\_split de 1 bit para indicar si la transformación se realiza con el mismo tamaño de la CU o no. En caso de que el indicador transform\_split se señale como verdadero, el residuo de la CU se divide a su vez en múltiples subbloques y la transformación se realiza en cada subbloque. En consecuencia, en un ejemplo, para al menos un nodo hoja de la estructura de árbol, el codificador de vídeo 22 puede incluir un elemento de sintaxis en el flujo de bits. En este ejemplo, el elemento de sintaxis que tiene un primer valor indica que una transformación que tiene el mismo tamaño que el bloque de vídeo correspondiente al nodo hoja se aplica a los datos residuales del bloque de vídeo correspondiente al nodo hoja; el elemento de sintaxis que tiene un segundo valor indica que múltiples transformaciones que tienen tamaños más pequeños que el bloque de vídeo correspondiente al nodo hoja se aplican a subbloques de los datos residuales del bloque de vídeo correspondiente al nodo hoja. En un ejemplo similar, para al menos un nodo hoja de la estructura de árbol, el decodificador de vídeo 30 puede obtener este elemento de sintaxis de un flujo de bits.

30 En algunos ejemplos, no se señala ningún indicador de división y solo se utiliza la transformación con un tamaño derivado para cuando hay residuos en la CU. Por ejemplo, para al menos un nodo hoja de la estructura de árbol, el codificador de vídeo 22 puede aplicar una misma transformación (por ejemplo, una transformada de coseno discreta, una transformada de seno discreta, etc.) a diferentes partes de datos residuales correspondientes al bloque de vídeo correspondiente al nodo hoja para convertir los datos residuales de un dominio de muestra a un dominio de transformación. En el dominio de muestra, los datos residuales se representan en términos de valores de muestras (por ejemplo, componentes de píxeles). En el dominio de transformación, los datos residuales pueden representarse en términos de coeficientes de frecuencia. Del mismo modo, para al menos un nodo hoja de la estructura de árbol, el decodificador de vídeo 30 puede aplicar una misma transformada (es decir, una transformada de coseno discreta inversa, una transformada de seno inversa, etc.) a diferentes partes de datos residuales correspondientes al bloque de vídeo correspondiente al nodo hoja para convertir los datos residuales de un dominio de transformación a un dominio de muestra.

45 En algunos ejemplos, para cada CU, si la CU no se divide en un árbol cuaternario cuadrado, o en un árbol binario simétrico, el tamaño de la transformación siempre se establece igual al tamaño del tamaño de la partición. Por ejemplo, para cada nodo no hoja respectivo de la estructura de árbol correspondiente a un bloque de vídeo particionado de acuerdo con un patrón de división de árbol cuaternario cuadrado o un patrón de división de árbol binario simétrico, los tamaños de transformación de las transformadas aplicadas a los datos residuales de los bloques de vídeo correspondientes a los nodos hijo del nodo no hoja respectivo siempre se establecen igual a los tamaños de los bloques de vídeo correspondientes a los nodos hijo del nodo no hoja respectivo.

La FIG. 11 es un diagrama de flujo que ilustra un ejemplo de operación de un codificador de vídeo, de acuerdo con otra técnica de ejemplo de esta divulgación. Uno o más elementos estructurales del codificador de vídeo 22, incluida la unidad de procesamiento de predicción 100, pueden configurarse para realizar las técnicas de la FIG. 11.

60 En un ejemplo de la divulgación, el codificador de vídeo 22 puede configurarse para recibir una imagen de los datos de vídeo (300), particionar la imagen de los datos de vídeo en una pluralidad de bloques utilizando tres o más estructuras de partición diferentes (302) y codificar la pluralidad de bloques de la imagen de los datos de vídeo (304). En un ejemplo de la divulgación, el codificador de vídeo 22 puede configurarse para particionar la imagen de los datos de vídeo en la pluralidad de bloques utilizando las tres o más estructuras de partición diferentes, en las que al menos tres de las tres o más estructuras de partición diferentes se pueden utilizar para al menos una profundidad de una estructura de árbol que representa cómo se particiona un bloque determinado de la imagen de los datos de vídeo. En un ejemplo, las tres o más estructuras de partición diferentes incluyen una estructura de partición de árbol terciario, y el codificador de vídeo 22 está configurado además para

particionar el bloque particular de los datos de vídeo utilizando un tipo de partición de árbol terciario de la estructura de partición de árbol terciario, en la que la estructura de partición de árbol terciario divide el bloque particular en tres subbloques sin dividir el bloque particular a través del centro del bloque particular, en el que un bloque central de los tres subbloques tiene un tamaño igual a la suma del tamaño de los otros dos de los tres subbloques, y en el que los otros dos de los tres subbloques tienen el mismo tamaño.

En otro ejemplo de la divulgación, las tres o más estructuras de partición diferentes incluyen además una estructura de partición de árbol cuaternario y una estructura de partición de árbol binario. En otro ejemplo de la divulgación, los tipos de partición de la estructura de partición de árbol cuaternario incluyen uno o más de un tipo de partición de árbol cuaternario cuadrado o un tipo de partición de árbol cuaternario rectangular, los tipos de partición de la estructura de partición de árbol binario incluyen uno o más de un tipo de partición de árbol binario simétrico o un tipo de partición de árbol binario no simétrico, y los tipos de partición de la estructura de partición de árbol terciario incluyen uno o más de un tipo de partición de árbol terciario simétrico o un tipo de partición de árbol terciario no simétrico.

En otro ejemplo de la divulgación, el codificador de vídeo 22 está configurado además para generar en un flujo de bits, elementos de sintaxis que indican una pluralidad de tipos de partición soportados de las tres o más estructuras de partición diferentes. En un ejemplo, la generación de los elementos de sintaxis a partir de un flujo de bits, incluida la generación de los elementos de sintaxis en uno o varios de los conjuntos de parámetros adaptables (APS), un conjunto de parámetros de secuencia (SPS), un conjunto de parámetros de imagen (PPS) o un encabezado de segmento.

En otro ejemplo de la divulgación, el codificador de vídeo 22 está configurado además para generar un elemento de sintaxis que indica que un bloque particular de la imagen de los datos de vídeo se particiona utilizando una estructura de partición de árbol terciario con un tipo de partición simétrica de árbol terciario, y particiona el bloque particular de la imagen de los datos de vídeo de tal manera que dos subbloques del bloque particular tengan el mismo tamaño.

En otro ejemplo de la divulgación, la pluralidad de bloques incluye un bloque particular que corresponde a un nodo hoja, y el codificador de vídeo 22 se configura además para generar un elemento de sintaxis en un flujo de bits, el elemento de sintaxis que tiene un primer valor que indica una transformación que tiene un mismo tamaño que el bloque particular de la imagen de los datos de vídeo correspondientes al nodo hoja se aplica a los datos residuales del bloque particular correspondiente al nodo hoja, el elemento de sintaxis que tiene un segundo valor que indica múltiples transformaciones que tienen tamaños más pequeños que el vídeo particular correspondiente al nodo hoja se aplica a los subbloques de los datos residuales del bloque particular correspondiente al nodo hoja, y se aplican una o más transformaciones a los datos residuales del bloque particular de los datos de vídeo de acuerdo con el elemento de sintaxis.

La FIG. 12 es un diagrama de flujo que ilustra un ejemplo de operación de un decodificador de vídeo, de acuerdo con otra técnica de ejemplo de esta divulgación. Uno o más elementos estructurales del decodificador de vídeo 30, incluida la unidad de decodificación de entropía 150 y/o la unidad de procesamiento de predicción 152, pueden configurarse para realizar las técnicas de la FIG. 12.

En un ejemplo de la divulgación, el decodificador de vídeo 30 está configurado para recibir un flujo de bits que incluye una secuencia de bits que forma una representación de una imagen codificada de los datos de vídeo (400), determinar una partición de la imagen codificada de los datos de vídeo en una pluralidad de bloques utilizando tres o más estructuras de partición diferentes (402), y reconstruir la pluralidad de bloques de la imagen codificada de los datos de vídeo (404). En un ejemplo, el decodificador de vídeo 30 está configurado para determinar la partición de la imagen codificada de los datos de vídeo en la pluralidad de bloques utilizando las tres o más estructuras de partición diferentes, en las que al menos tres de las tres o más estructuras de partición diferentes se pueden utilizar para al menos una profundidad de una estructura de árbol que representa cómo se particiona un bloque determinado de la imagen codificada de los datos de vídeo. En un ejemplo, las tres o más estructuras de partición diferentes incluyen una estructura de partición de árbol terciario, y el decodificador de vídeo 30 se configura además para determinar la partición del bloque particular de los datos de vídeo utilizando un tipo de partición de árbol terciario de la estructura de partición de árbol terciario, en la que la estructura de partición de árbol terciario divide el bloque particular en tres subbloques sin dividir el bloque particular a través del centro del bloque particular, en el que un bloque central de los tres subbloques tiene un tamaño igual a la suma de un tamaño de los otros dos de los tres subbloques, y en el que los otros dos de los tres subbloques tienen el mismo tamaño.

En otro ejemplo de la divulgación, las tres o más estructuras de partición diferentes incluyen además una estructura de partición de árbol cuaternario y una estructura de partición de árbol binario. En otro ejemplo, los tipos de partición de la estructura de partición de árbol cuaternario incluyen uno o más de un tipo de partición de árbol cuaternario cuadrado o un tipo de partición de árbol cuaternario rectangular, los tipos de partición de la estructura de partición de árbol binario incluyen uno o más de un tipo de partición de árbol binario simétrico o un tipo de partición de árbol binario no simétrico, y los tipos de partición de la estructura de partición de árbol



terciario incluyen uno o más de un tipo de partición de árbol terciario simétrico o un tipo de partición de árbol terciario no simétrico.

En otro ejemplo de la divulgación, el decodificador de vídeo 30 está configurado además para recibir, desde el flujo de bits, elementos de sintaxis que indican una pluralidad de tipos de partición soportados de las tres o más estructuras de partición diferentes, y determinar la partición de la imagen codificada de los datos de vídeo basado en el elemento de sintaxis recibido. En otro ejemplo de la divulgación, el decodificador de vídeo 30 se configura además para recibir los elementos de sintaxis del flujo de bits, incluida la recepción de los elementos de sintaxis en uno o más de los conjuntos de parámetros adaptables (APS), un conjunto de parámetros de secuencia (SPS), un conjunto de parámetros de imagen (PPS) o un encabezado de segmento.

En otro ejemplo de la divulgación, el decodificador de vídeo 30 está configurado además para recibir un elemento de sintaxis que indica que un bloque particular de la imagen codificada de los datos de vídeo se particiona utilizando una estructura de partición de árbol terciario con un tipo de partición de árbol terciario simétrico, y determinar una partición del bloque particular de la imagen codificada de los datos de vídeo de modo que dos subbloques del bloque particular tengan el mismo tamaño.

En otro ejemplo de la divulgación, la pluralidad de bloques incluye un bloque particular que corresponde a un nodo hoja, y el decodificador de vídeo 30 está configurado además para recibir un elemento de sintaxis del flujo de bits, el elemento de sintaxis que tiene un primer valor que indica una transformación que tiene un mismo tamaño que el bloque particular de la imagen codificada de los datos de vídeo correspondientes al nodo hoja se aplica a los datos residuales del bloque particular correspondientes al nodo hoja, el elemento de sintaxis que tiene un segundo valor que indica múltiples transformaciones que tienen tamaños más pequeños que el bloque particular correspondiente al nodo hoja se aplican a los subbloques de los datos residuales del bloque particular correspondiente al nodo hoja, y aplican una o más transformaciones al bloque particular de datos de vídeo de acuerdo con el elemento de sintaxis.

Ciertos aspectos de esta divulgación han sido descritos con respecto a las extensiones del estándar HEVC con fines ilustrativos. Sin embargo, las técnicas descritas en esta divulgación pueden ser útiles para otros procesos de codificación de vídeo, incluidos otros procesos de codificación de vídeo estándar o propietario que aún no se han desarrollado.

Un codificador de vídeo, tal como se describe en esta divulgación, puede referirse a un codificador de vídeo o a un decodificador de vídeo. Similarmente, una unidad de codificación de vídeo puede referirse a un codificador de vídeo o a un decodificador de vídeo. Del mismo modo, la codificación de vídeo puede referirse a la codificación de vídeo o a la decodificación de vídeo, según corresponda. En esta divulgación, la frase "basado en" puede indicar basado solo en, basado al menos en parte en, o basado de alguna manera en. Esta divulgación puede utilizar el término "unidad de vídeo" o "bloque de vídeo" o "bloque" para referirse a uno o más bloques de muestra y estructuras de sintaxis utilizadas para codificar muestras del uno o más bloques de muestras. Algunos tipos de ejemplos de unidades de vídeo pueden incluir las CTU, las CU, las PU, las unidades de transformación (TUs), los macrobloques, las particiones de macrobloques, y así sucesivamente. En algunos contextos, la discusión de las PUs puede intercambiarse con la discusión de macrobloques o particiones de macrobloques. Algunos ejemplos de tipos de bloques de vídeo pueden incluir bloques de árbol de codificación, bloques de codificación y otros tipos de bloques de datos de vídeo.

Debe reconocerse que, dependiendo del ejemplo, ciertos actos o eventos de cualquiera de las técnicas descritas en este documento pueden realizarse en una secuencia diferente, pueden agregarse, fusionarse u omitirse por completo (por ejemplo, no todos los actos o eventos descritos son necesarios para la práctica de las técnicas). Además, en ciertos ejemplos, los actos o eventos pueden realizarse simultáneamente, por ejemplo, a través del procesamiento de múltiples hilos, el procesamiento de interrupciones o múltiples procesadores, en lugar de secuencialmente.

En uno o más ejemplos, las funciones descritas pueden ser implementadas en hardware, software, firmware, o cualquier combinación de los mismos. Si se implementan en software, las funciones pueden almacenarse o transmitirse como una o más instrucciones o códigos en un medio legible por ordenador y ser ejecutadas por una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible como los medios de almacenamiento de datos, o medios de comunicación, incluido cualquier medio que facilite la transferencia de un programa informático de un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador generalmente pueden corresponder a (1) medios de almacenamiento tangibles legibles por ordenador que no son transitorios o (2) un medio de comunicación como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que puedan acceder uno o más ordenadores o uno o más procesadores para recuperar instrucciones, 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 limitativo, 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 pueda utilizarse para almacenar el código de programa deseado en forma de instrucciones o estructuras de datos y al que pueda acceder un ordenador. Además, cualquier conexión se denomina correctamente un medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, un servidor u otra fuente remota utilizando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de suscriptor digital (DSL) o tecnologías inalámbricas como infrarrojos, radio y microondas, el cable coaxial, el cable de fibra óptica, el par trenzado, el DSL o las tecnologías inalámbricas como el infrarrojo, la radio y el microondas se incluyen en la definición de medio. Sin embargo, debe entenderse que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que se dirigen a medios de almacenamiento tangibles no transitorios. El disco magnético y el disco óptico, tal como se utilizan en este documento, incluyen el disco compacto (CD), el disco láser, el disco óptico, el disco versátil digital (DVD), el disquete y el disco Blu-ray, donde los discos magnéticos suelen reproducir datos magnéticamente, mientras que los discos ópticos reproducen datos ópticamente con láseres. Las combinaciones de lo anterior también deben incluirse dentro del alcance de los medios legibles por ordenador.

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

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

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

## REIVINDICACIONES

1. Un método de decodificación de datos de vídeo, el método que comprende:

5 recibir (400) un flujo de bits que incluye una secuencia de bits que forma una representación de una imagen codificada de los datos de vídeo;

10 determinar (402) una partición de la imagen codificada de los datos de vídeo en una pluralidad de bloques utilizando tres estructuras de partición diferentes, en la que la partición se selecciona entre tres estructuras de partición diferentes para al menos una profundidad de una estructura de árbol que representa cómo se particiona un bloque particular de la imagen codificada de los datos de vídeo, y en el que las tres estructuras de partición diferentes incluyen una estructura de partición de árbol terciario, una estructura de partición de árbol cuaternario y una estructura de partición de árbol binario;

15 determinar la partición de al menos un bloque de la imagen codificada de los datos de vídeo utilizando un tipo de partición de árbol terciario de la estructura de partición de árbol terciario, en la que la estructura de partición de árbol terciario divide el al menos un bloque en tres subbloques sin dividir el al menos un bloque a través de la línea central del al menos un bloque, en el que al menos uno de los tres subbloques es un nodo no hoja de la estructura de árbol; y

20 reconstruir (404) la pluralidad de bloques de la imagen codificada de los datos de vídeo

2. El método de la reivindicación 1, que comprende además:

25 recibir, del flujo de bits, elementos de sintaxis que indican una pluralidad de tipos de partición soportados de las tres estructuras de partición diferentes; y

30 determinar el particionamiento de la imagen codificada de los datos de vídeo basados en los elementos de sintaxis recibidos.

3. El método de la reivindicación 2, en el que la recepción de los elementos de sintaxis comprende la recepción de los elementos de sintaxis del flujo de bits, incluida la recepción de los elementos de sintaxis en uno o más de un conjunto de parámetros adaptativos, APS, un conjunto de parámetros de secuencia, SPS, un conjunto de parámetros de imagen, PPS o un encabezado de segmento.

35 4. El método de la reivindicación 1, el método que comprende además:

40 recibir un elemento de sintaxis que indica que el al menos un bloque de la imagen codificada de los datos de vídeo está particionado utilizando la estructura de partición de árbol terciario con un tipo de partición simétrica de árbol terciario; y

determinar el particionamiento del al menos un bloque de la imagen codificada de los datos de vídeo de modo que dos subbloques del al menos un bloque tengan el mismo tamaño.

45 5. El método de la reivindicación 1, en el que la pluralidad de bloques incluye un bloque correspondiente a un nodo hoja, el método que comprende además:

50 recibir un elemento de sintaxis del flujo de bits, el elemento de sintaxis que tiene un primer valor que indica una transformación que tiene el mismo tamaño que el bloque correspondiente al nodo hoja se aplica a los datos residuales del bloque correspondiente al nodo hoja, el elemento de sintaxis que tiene un segundo valor que indica múltiples transformaciones que tienen tamaños más pequeños que el bloque correspondiente al nodo hoja se aplica a los subbloques de los datos residuales del bloque correspondiente al nodo hoja; y

55 aplicar una o más transformaciones al bloque correspondiente al nodo hoja de acuerdo con el elemento de sintaxis.

6. El método de la reivindicación 1, en el que un bloque central de los tres subbloques tiene un tamaño igual a la suma de un tamaño de dos de los tres subbloques.

60 7. Un método de codificación de datos de vídeo, que comprende:

recibir (300) una imagen de los datos de vídeo;

65 particionar (302) la imagen de los datos de vídeo en una pluralidad de bloques utilizando tres estructuras de partición diferentes, en las que la partición se selecciona entre tres estructuras de partición diferentes para al menos una profundidad de una estructura de árbol que representa cómo se particiona un bloque particular de

la imagen de los datos de vídeo, y en la que las tres estructuras de partición diferentes incluyen una estructura de partición de árbol terciario, una estructura de partición de árbol cuaternario y una estructura de partición de árbol binario;

5      particionar al menos un bloque de la imagen de los datos de vídeo utilizando un tipo de partición de árbol terciario de la estructura de partición de árbol terciario, en la que la estructura de partición de árbol terciario divide el al menos un bloque en tres subbloques sin dividir el al menos un bloque a través de la línea central del al menos un bloque, en el que al menos uno de los tres subbloques es un nodo no hoja de la estructura de árbol; y

10     codificar (304) la pluralidad de bloques de la imagen de los datos de vídeo

8. El método de la reivindicación 1 o de la reivindicación 7, en el que un bloque central de los tres subbloques tiene un tamaño igual a la suma de un tamaño de los otros dos de los tres subbloques, y en el que los otros dos de los tres subbloques tienen el mismo tamaño.

9. El método de la reivindicación 1 o de la reivindicación 7, en el que los tipos de partición de la estructura de partición de árbol cuaternario incluyen uno o más de un tipo de partición de árbol cuaternario cuadrado o un tipo de partición de árbol cuaternario rectangular,

20     en el que los tipos de partición de la estructura de partición de árbol binario incluyen uno o más de un tipo de partición de árbol binario simétrico o de un tipo de partición de árbol binario no simétrico,

25     en el que, los tipos de partición para la estructura de partición de árbol terciario incluyen uno o más de un tipo de partición de árbol terciario simétrico o un tipo de partición de árbol terciario no simétrico.

10. El método de la reivindicación 7, que comprende además:

30     generar, en un flujo de bits, elementos de sintaxis que indican una pluralidad de tipos de partición soportados de las tres estructuras de partición diferentes.

11. El método de la reivindicación 10, en el que la generación de los elementos de sintaxis comprende la generación de los elementos de sintaxis a partir de un flujo de bits, incluida la generación de los elementos de sintaxis en uno o más de un conjunto de parámetros adaptativos, APS, un conjunto de parámetros de secuencia, SPS, un conjunto de parámetros de imagen, PPS o un encabezado de segmento.

12. El método de la reivindicación 7, el método que comprende además:

40     generar un elemento de sintaxis que indique que al menos un bloque de la imagen de los datos de vídeo está particionado utilizando la estructura de partición de árbol terciario con un tipo de partición de árbol terciario simétrico; y

45     particionar al menos un bloque de la imagen de los datos de vídeo de modo que dos subbloques del bloque en particular tengan el mismo tamaño.

13. El método de la reivindicación 7, en el que la pluralidad de bloques incluye un bloque correspondiente a un nodo hoja, que comprende además:

50     generar un elemento de sintaxis en un flujo de bits, el elemento de sintaxis que tiene un primer valor que indica una transformación que tiene el mismo tamaño que el bloque correspondiente al nodo hoja se aplica a los datos residuales del bloque correspondiente al nodo hoja, el elemento de sintaxis que tiene un segundo valor que indica múltiples transformaciones que tienen tamaños más pequeños que el bloque correspondiente al nodo hoja se aplican a los subbloques de los datos residuales del bloque correspondiente al nodo hoja; y

55     aplicar una o más transformaciones a los datos residuales del bloque correspondiente al nodo hoja de acuerdo con el elemento de sintaxis.

14. Un aparato configurado para decodificar datos de vídeo, el aparato que comprende:

60     una memoria configurada para almacenar los datos de vídeo; y

   circuitos de decodificación de vídeo configurados para:

65     recibir un flujo de bits que incluya una secuencia de bits que forme una representación de una imagen codificada de los datos de vídeo;

determinar una partición de la imagen codificada de los datos de vídeo en una pluralidad de bloques utilizando tres estructuras de partición diferentes, en las que la partición se selecciona entre tres estructuras de partición diferentes para al menos una profundidad de una estructura de árbol que representa cómo se particiona un bloque particular de la imagen codificada de los datos de vídeo, y en el que las tres estructuras de partición diferentes incluyen una estructura de partición de árbol terciario, una estructura de partición de árbol cuaternario y una estructura de partición de árbol binario;

determinar la partición de al menos un bloque de la imagen codificada de los datos de vídeo utilizando un tipo de partición de árbol terciario de la estructura de partición de árbol terciario, en la que la estructura de partición de árbol terciario divide el al menos un bloque en tres subbloques sin dividir al menos un bloque a través de la línea central del al menos un bloque, en el que al menos uno de los tres subbloques es un nodo no hoja de la estructura de árbol; y

reconstruir la pluralidad de bloques de la imagen codificada de los datos de vídeo.

15. Un aparato configurado para decodificar datos de vídeo, que comprende:

medios para recibir un flujo de bits que incluya una secuencia de bits que forme una representación de una imagen codificada de los datos de vídeo;

medios para determinar una partición de la imagen codificada de los datos de vídeo en una pluralidad de bloques utilizando tres estructuras de partición diferentes, en las que la partición se selecciona entre tres estructuras de partición diferentes para al menos una profundidad de una estructura de árbol que representa cómo se particiona un bloque particular de la imagen codificada de los datos de vídeo, y en el que las tres estructuras de partición diferentes incluyen una estructura de partición de árbol terciario, una estructura de partición de árbol cuaternario y una estructura de partición de árbol binario;

medios para determinar una partición de al menos un bloque de la imagen codificada de los datos de vídeo utilizando un tipo de partición de árbol terciario de la estructura de partición de árbol terciario, en la que la estructura de partición de árbol terciario divide al menos un bloque en tres subbloques sin dividir el al menos un bloque a través de la línea central del al menos un bloque, en el que al menos uno de los tres subbloques es un nodo no hoja de la estructura de árbol; y

medios para reconstruir la pluralidad de bloques de la imagen codificada de los datos de vídeo.

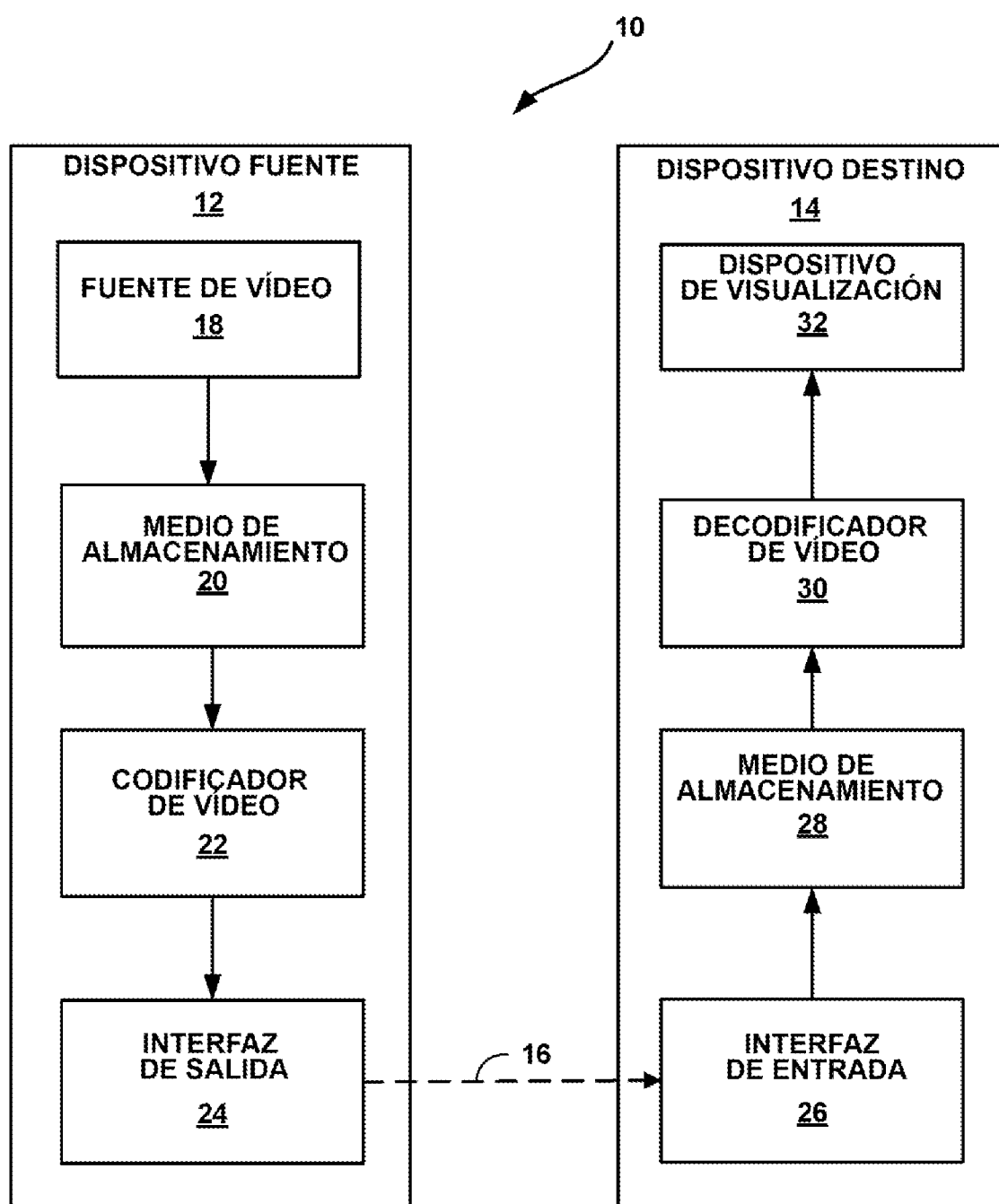
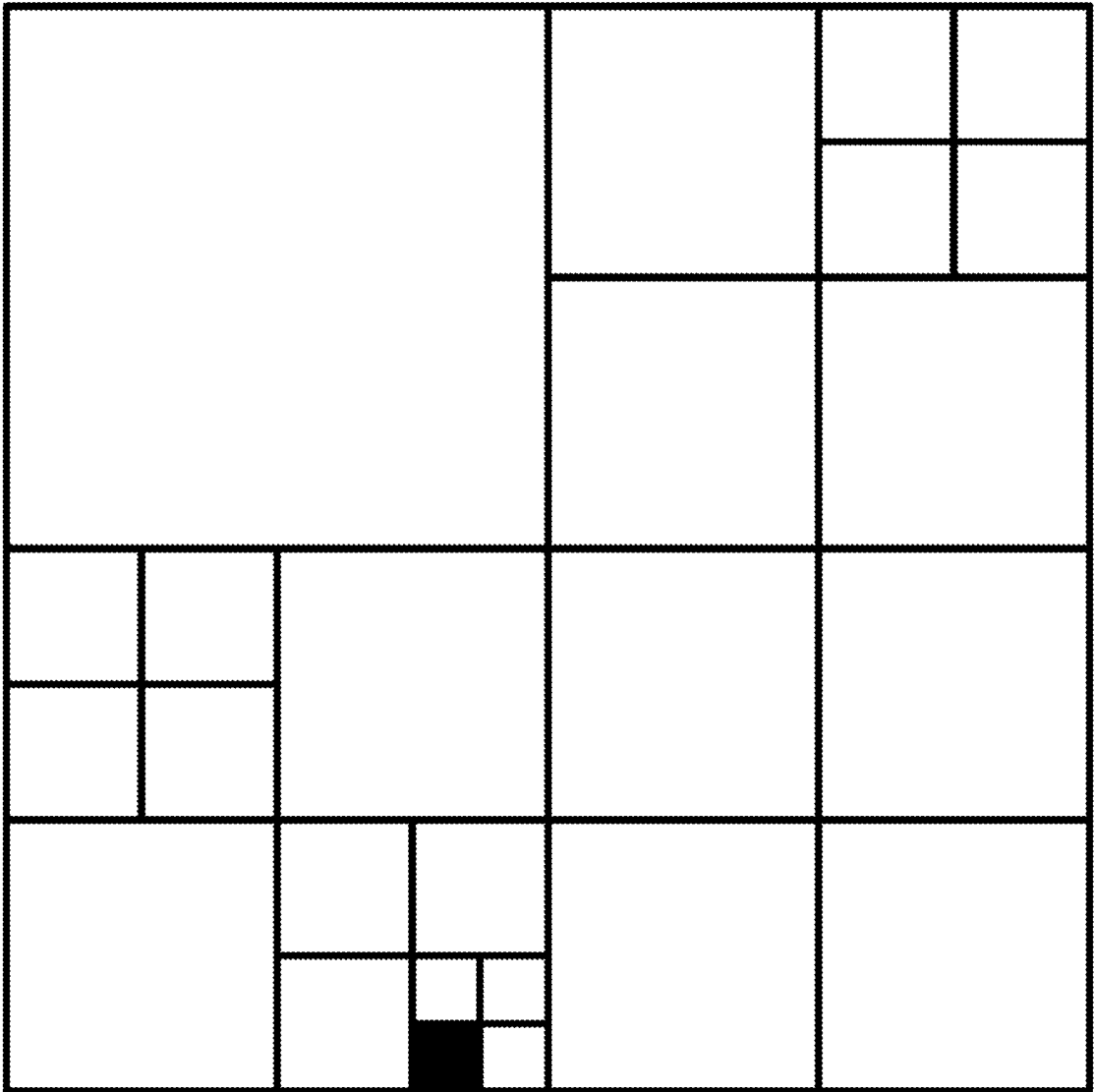


FIG. 1



**FIG. 2**

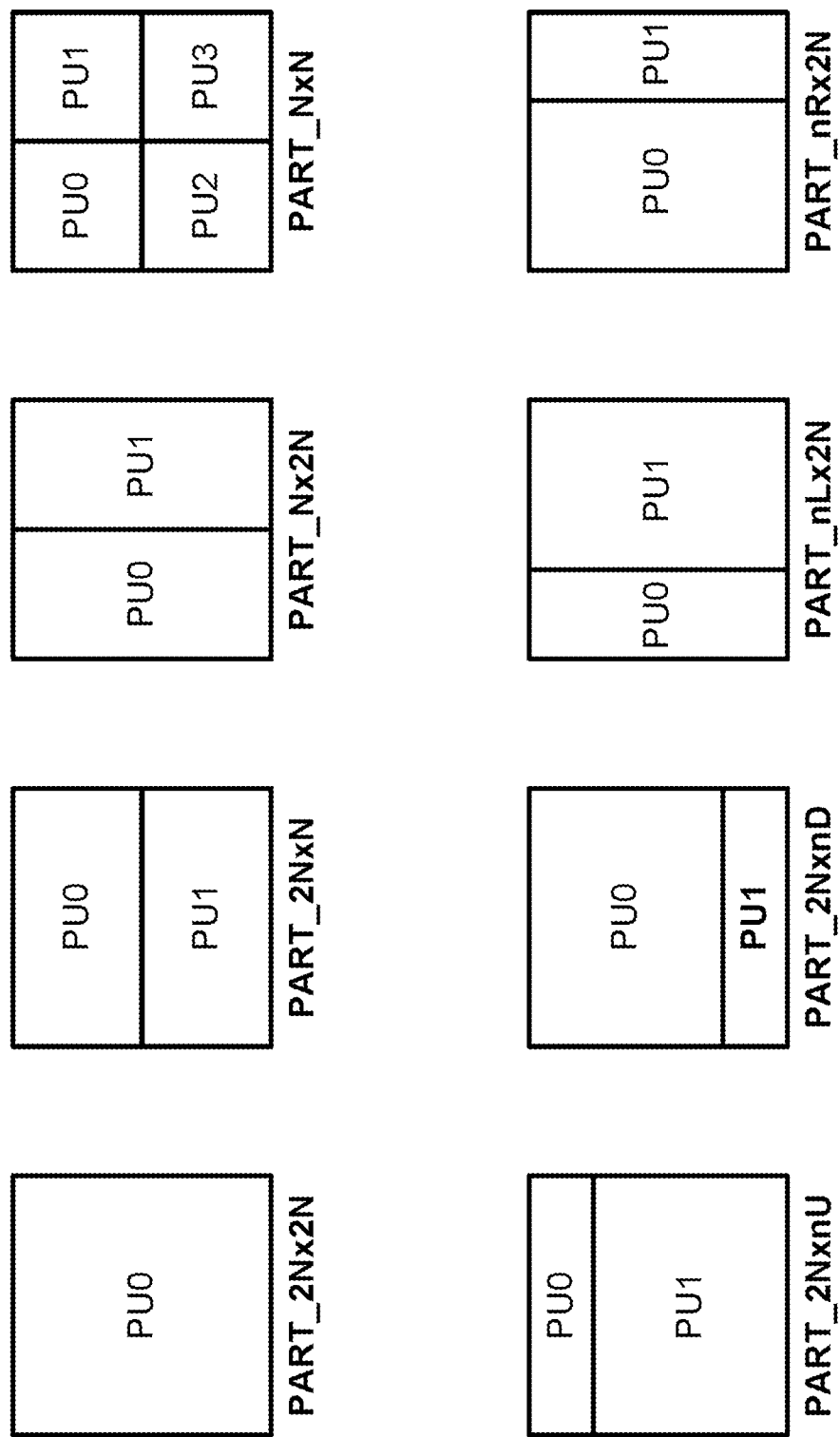


FIG. 3



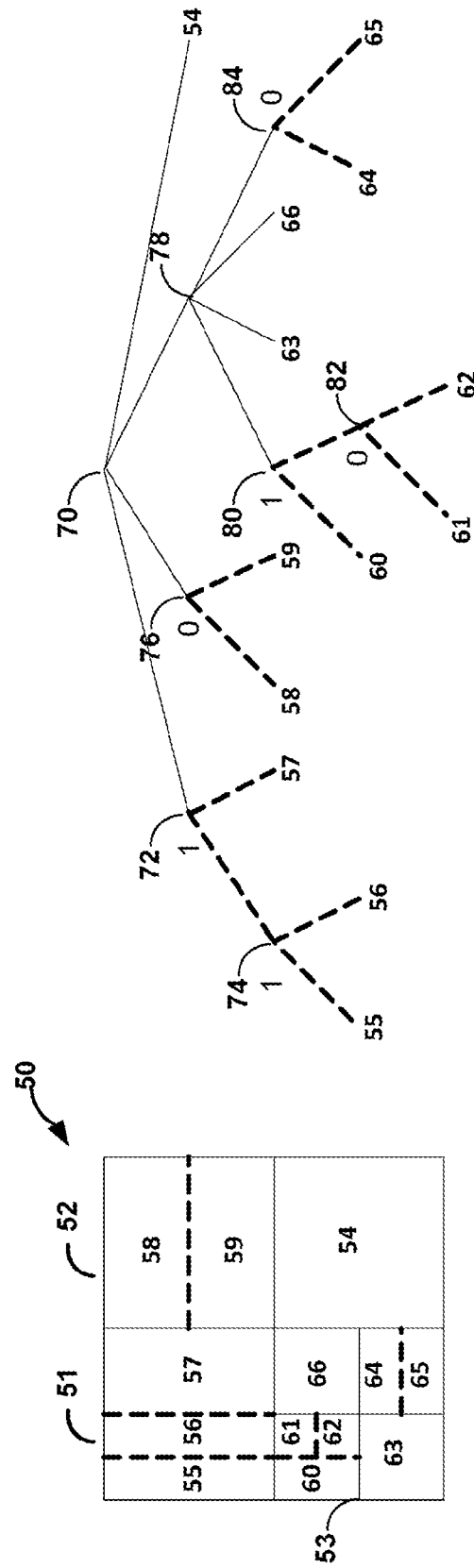
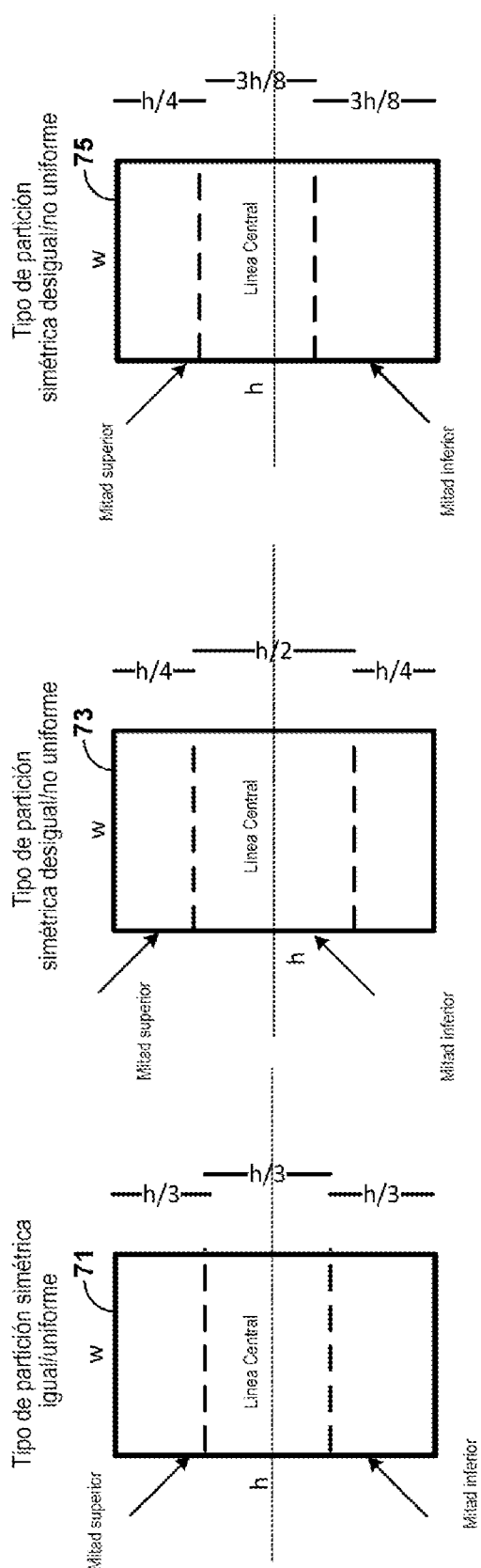


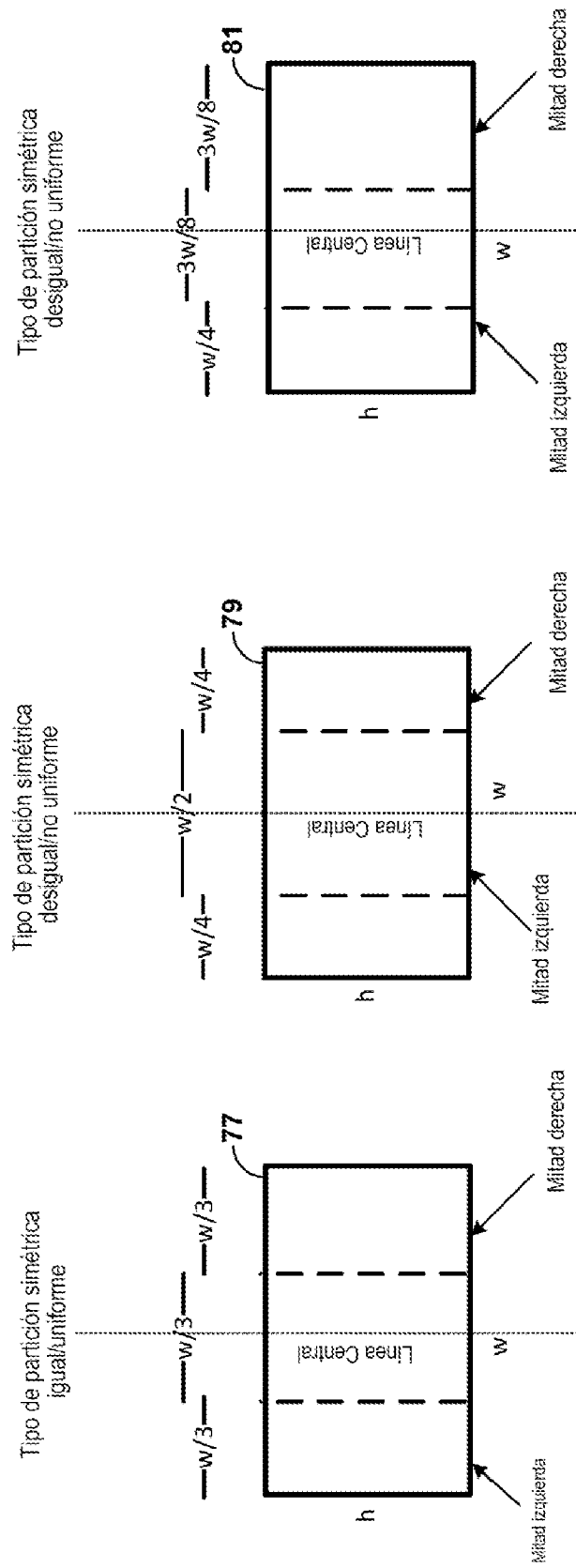
FIG. 4A

FIG. 4B



## TIPOS DE PARTICIÓN HORIZONTAL DE ÁRBOL TERCARIO

**FIG. 5A**



**FIG. 5B**

TIPOS DE PARTICIÓN VERTICAL DE ÁRBOL TERCARIO

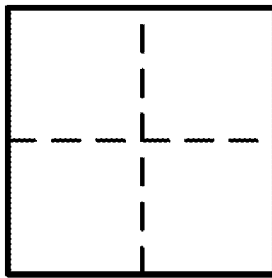


FIG. 6A

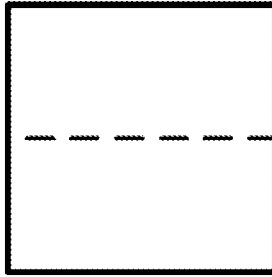


FIG. 6B

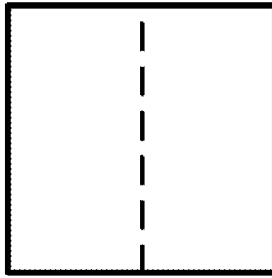


FIG. 6C

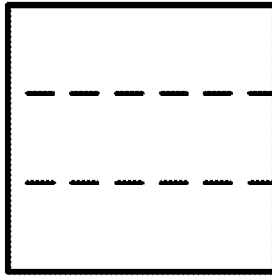


FIG. 6D

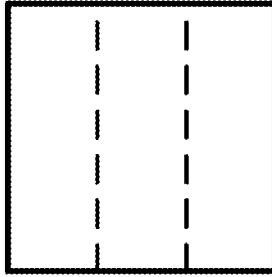


FIG. 6E

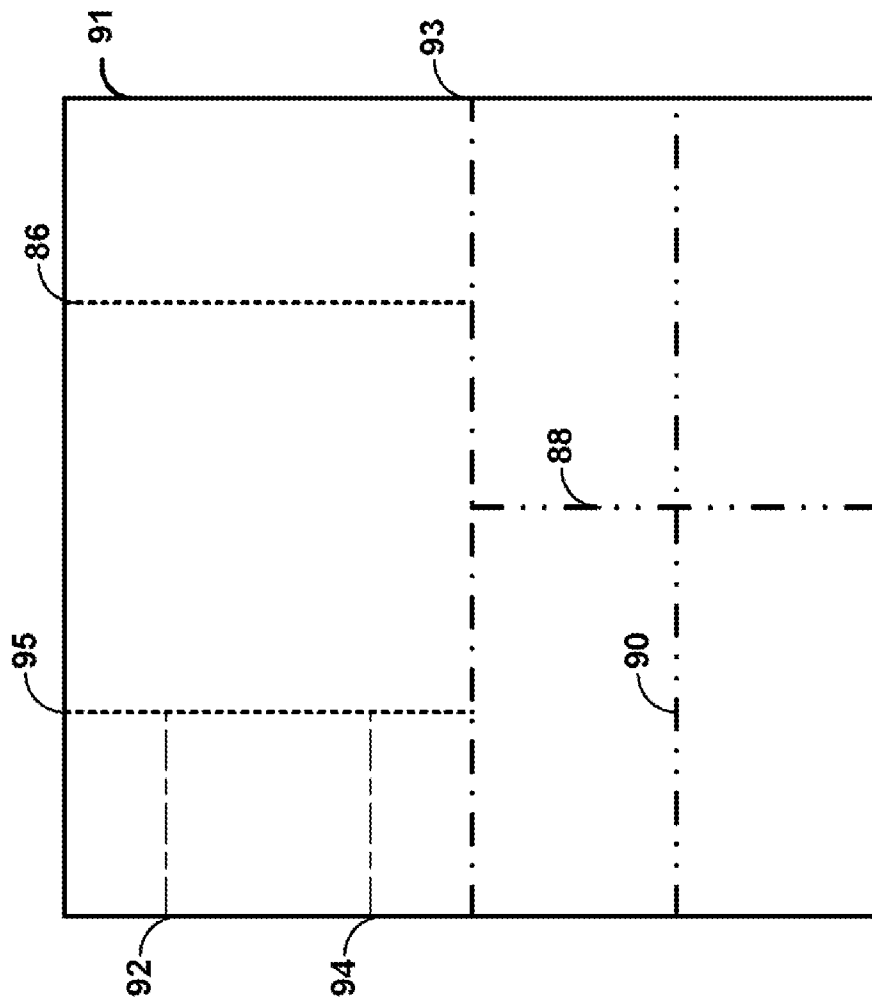


FIG. 7

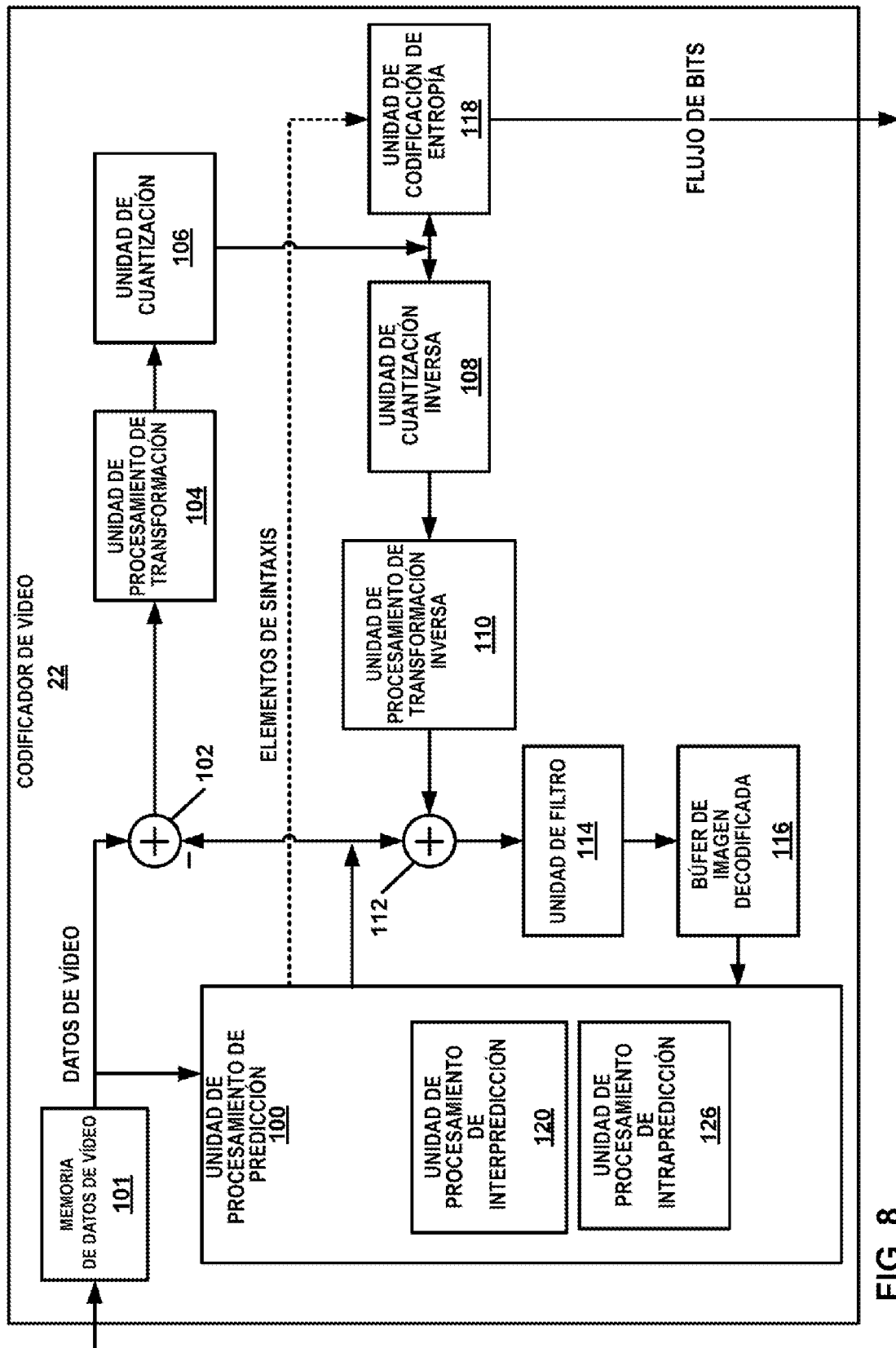


FIG. 8

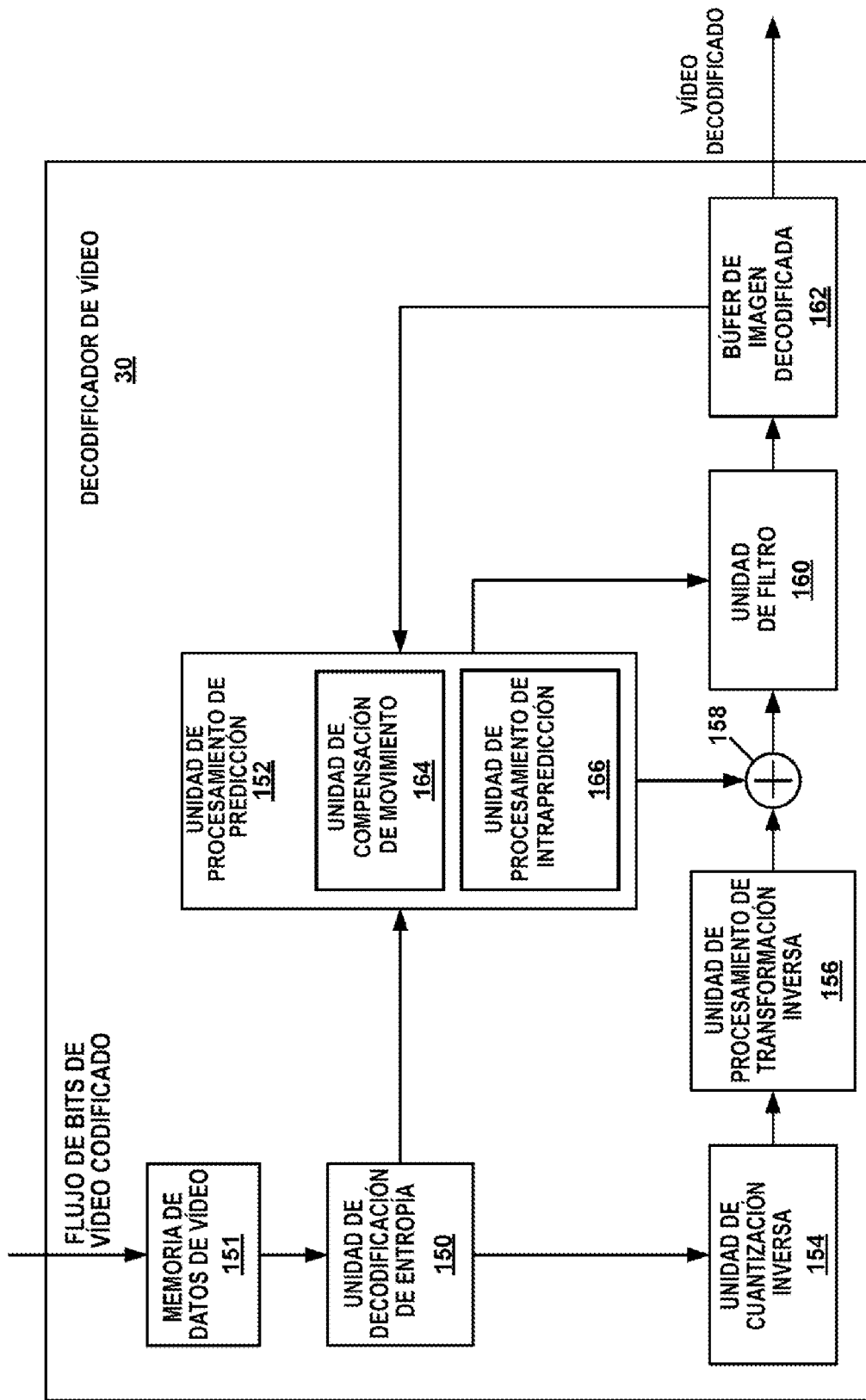
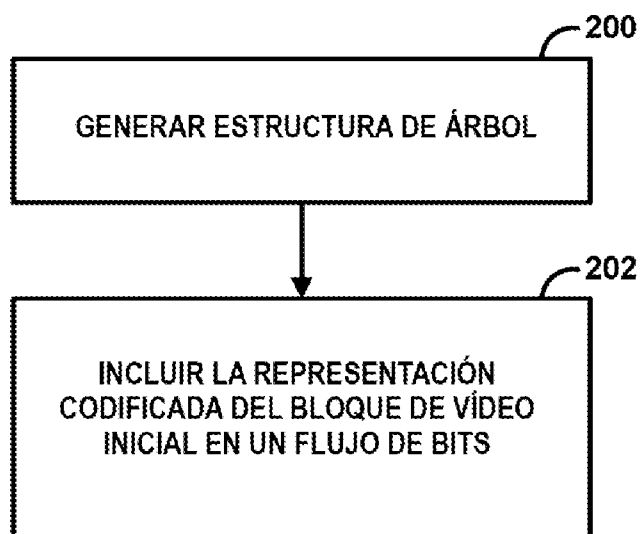
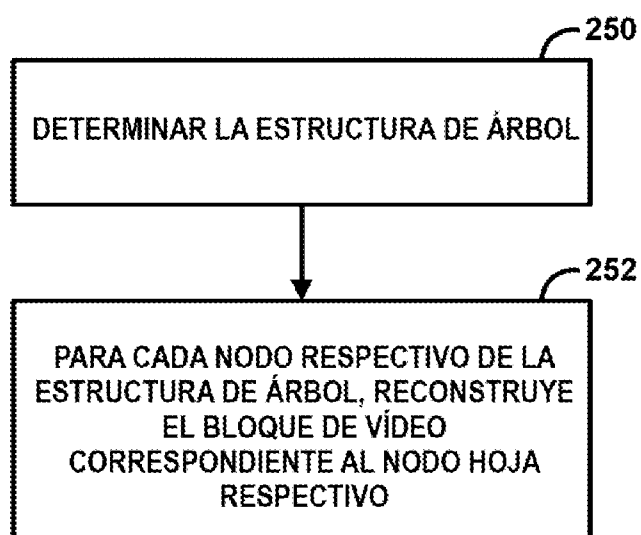


FIG. 9

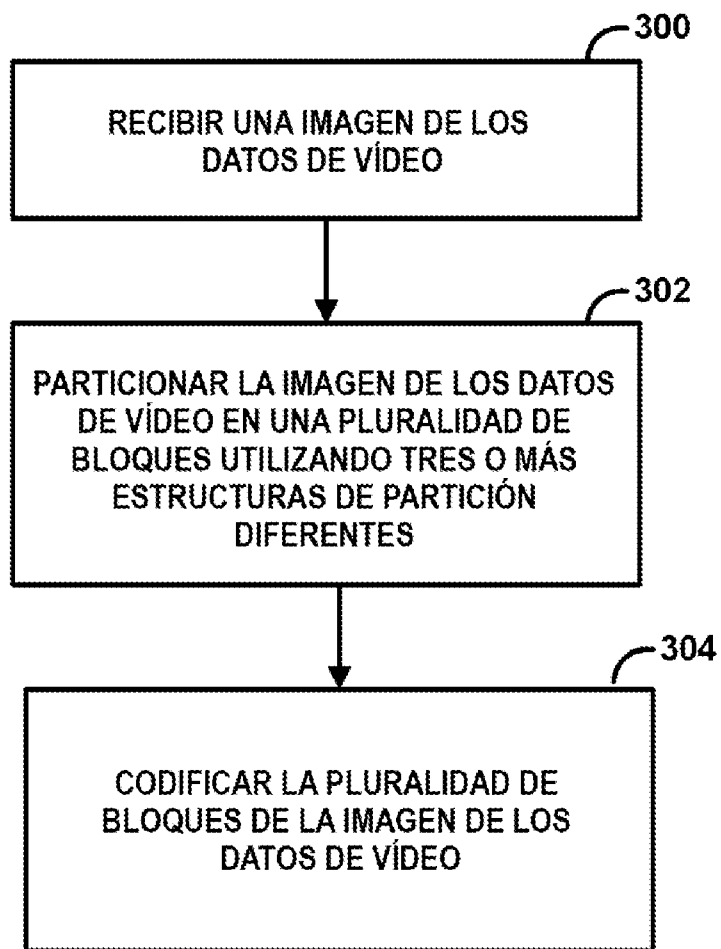


**FIG. 10A**



**FIG. 10B**





**FIG. 11**

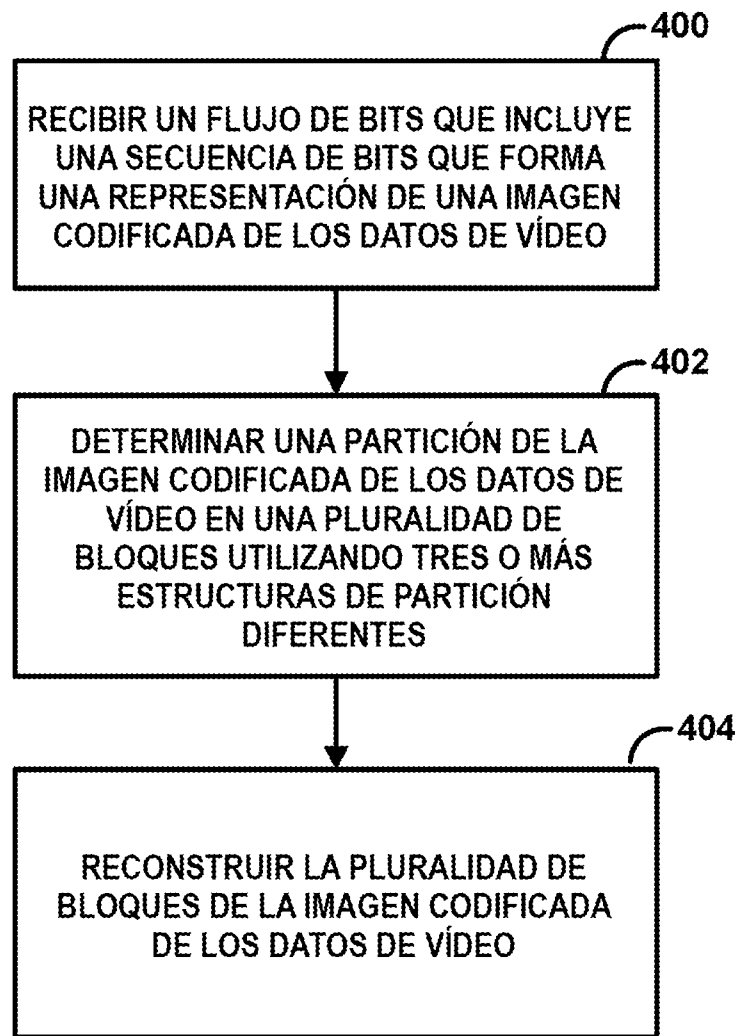


FIG. 12