

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **3 022 200**

51 Int. Cl.:

H04N 19/176 (2014.01)
H04N 19/119 (2014.01)
H04N 19/134 (2014.01)
H04N 19/96 (2014.01)
H04N 19/593 (2014.01)
H04N 19/463 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **07.09.2017** **PCT/US2017/050464**

87 Fecha y número de publicación internacional: **15.03.2018** **WO18049020**

96 Fecha de presentación y número de la solicitud europea: **07.09.2017** **E 17771939 (0)**

97 Fecha y número de publicación de la concesión europea: **02.04.2025** **EP 3510776**

54 Título: **Codificación de tipo de árbol para codificación de vídeo**

30 Prioridad:

07.09.2016 US 201662384585 P
06.09.2017 US 201715697134

45 Fecha de publicación y mención en BOPI de la
traducción de la patente:
28.05.2025

73 Titular/es:

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

72 Inventor/es:

CHUANG, HSIAO-CHIANG;
LI, XIANG;
CHEN, JIANLE;
ZOU, FENG;
CHIEN, WEI-JUNG;
CHEN, YI-WEN y
KARCZEWICZ, MARTA

74 Agente/Representante:

ISERN JARA, Jorge

ES 3 022 200 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

Codificación de tipo de árbol para codificación de vídeo

5 CAMPO TÉCNICO

La presente descripción se refiere a la codificación de vídeo.

10 ANTECEDENTES

Las capacidades de vídeo digital se pueden incorporar en una amplia gama de dispositivos, incluidos televisores digitales, sistemas de difusión directa digital, sistemas de difusión inalámbrica, asistentes digitales personales (Personal Digital Assistants, PDA), ordenadores portátiles o de escritorio, tabletas, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos celulares o de radio por satélite, los llamados "teléfonos inteligentes", dispositivos de videoconferencia, dispositivos de transmisión de vídeo y similares. Los dispositivos de vídeo digital implementan técnicas de codificación de vídeo, tales como las descritas en las normas definidas por diversas normas de codificación de vídeo. Las normas de codificación de vídeo incluyen 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 conocida como ISO/IEC MPEG-4 AVC), incluyendo las extensiones de codificación de vídeo escalable (Scalable Video Coding, SVC) y codificación de vídeo multivista (Multi-view Video Coding, MVC). Además, el equipo de colaboración conjunta sobre codificación de vídeo (Joint Collaboration Team on Video Coding, JCT-VC) del grupo de expertos en codificación de vídeo (Video Coding Experts Group, VCEG) de la UIT-T y el grupo de expertos en imágenes en movimiento (Motion Picture Experts Group, MPEG) de la ISO/IEC ha desarrollado recientemente un nuevo estándar de codificación de vídeo, denominado codificación de vídeo de alta eficiencia (High Efficiency Video Coding, HEVC). El borrador más reciente de la especificación HEVC, y referido en lo sucesivo como "HEVC WD", está disponible en http://phenix.int-evry.fr/jct/doc_end_user/documents/14_Vienna/wg11/JCTVC-N1003-v1.zip. La especificación de HEVC y sus extensiones que incluyen las extensiones de rango de formato (Format Range, RExt), escalabilidad (SHVC) y vistas múltiples (MV-HEVC) y las extensiones de contenido de pantalla (Screen Content) están disponibles en http://phenix.int-evry.fr/jct/doc_end_user/current_document.php?id=10481. Actualmente, las normas ITU-T VCEG (Q6/16) e ISO/IEC MPEG (JTC 1/SC 29/WG 11) están estudiando la posible necesidad de normalización de la futura tecnología de codificación de vídeo con una capacidad de compresión que supere significativamente a la de la norma HEVC actual (incluyendo sus extensiones actuales y a corto plazo para codificación de contenido de pantalla y codificación de alto rango dinámico). Los grupos están trabajando juntos en esta actividad de exploración en un esfuerzo conjunto de colaboración conocido como equipo Conjunto de Exploración de Vídeo (Joint Video Exploration Team, JVET) para evaluar los diseños de tecnología de compresión propuestos por sus expertos en esta área. El JVET se reunió por primera vez durante el 19-21 de octubre de 2015. La última versión del software de referencia, es decir, Joint Exploration Model 7 (JEM 7), se puede descargar de: https://jvet.hhi.fraunhofer.de/svn/svn_HMEMSoftware/tags/HM-16.6-EM-7.0/. La descripción del algoritmo para JEM-7.0 se describe además en "Algorithm description of Joint Exploration Test Model 7", de J. Chen, E. Alshina, G. J. Sullivan, J.-R. Ohm, J. Boyce, JVET-G1001, Ginebra, julio de 2017.

Los dispositivos de vídeo pueden transmitir, recibir, codificar, decodificar y/o almacenar información de vídeo digital de manera más eficiente mediante la implementación de dichas técnicas de codificación de vídeo. Las técnicas de codificación de vídeo incluyen predicción espacial (intra-imagen) y/o predicción temporal (inter-imagen) para reducir o eliminar la redundancia inherente a las secuencias de vídeo. Para la codificación de vídeo basada en bloques, un fragmento de vídeo (por ejemplo, una trama de vídeo o una parte de una trama de vídeo) se puede dividir en bloques de vídeo, que para algunas técnicas también pueden denominarse bloques de árbol, unidades de codificación (Coding Unit, CU) y/o nodos de codificación. Los bloques de vídeo en un fragmento intracodificado (I) de una imagen se codifican utilizando predicción espacial con respecto a las muestras de referencia en bloques vecinos en la misma imagen. Los bloques de vídeo en un fragmento intercodificado (P o B) de una imagen pueden usar predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen o predicción temporal con respecto a muestras de referencia en otras imágenes de referencia. Las imágenes se pueden denominar tramas y las imágenes de referencia se pueden denominar tramas de referencia.

La predicción espacial o temporal da como resultado un bloque predictivo para un bloque que se va a codificar. Los datos residuales representan diferencias de píxeles entre el bloque original para codificar y el bloque predictivo. Un bloque intercodificado se codifica según un vector de movimiento que apunta a un bloque de muestras de referencia que forma el bloque predictivo, y los datos residuales que indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intracodificado se codifica según un modo de intracodificación y los datos residuales. Para una compresión adicional, los datos residuales se pueden transformar del dominio de píxeles a un dominio de transformada, lo que da como resultado coeficientes de transformada residuales, que, a continuación, se pueden cuantificar. Los coeficientes de transformada cuantificados, inicialmente dispuestos en una matriz bidimensional, pueden someterse a un barrido para producir un vector unidimensional de coeficientes de transformada, y se puede aplicar la codificación de entropía para lograr aún más compresión.

El documento WO 2014/120367 se refiere a codificación de datos de vídeo, e indica que los datos de vídeo pueden dividirse, por ejemplo usando técnicas de división de árbol binario o de división de árbol k-d. Se describe que la técnica de división utilizada puede determinarse basándose en el tipo de imagen de las tramas individuales o en una característica de la trama que se está dividiendo.

"Fast prediction unit partition mode selection for high-efficiency video coding intercoding using motion homogeneous coding units" de Wu y col., Journal of Electronic Imaging, SPIE/IS&T, 1000 20th St, Bellingham, WA, 98255-6705, USA, Vol. 24, nº 6, noviembre de 2015, páginas 063024-1 a 063024-21 enseña la reducción en la complejidad computacional de los sistemas de codificación de vídeo mediante el intercambio con la eficiencia de codificación, en particular a través del uso de un esquema de selección de modo de división de unidades de predicción rápida en el que el número de modos de división de unidades de predicción candidatos se reduce. En un ejemplo en particular, se hace referencia al uso de un máximo de dos modos de predicción para cada nivel de profundidad de la unidad actual.

RESUMEN

En general, la presente descripción describe técnicas relacionadas con la codificación (p. ej., codificación y/o decodificación) de datos de vídeo. Algunos aspectos de esta descripción se dirigen a mejorar la eficiencia de la codificación y señalización con respecto a estructuras de división de bloques que se han desarrollado con posterioridad a HEVC.

En un ejemplo, esta descripción se dirige a un procedimiento de codificación de datos de vídeo como se define en la reivindicación 1, con una característica preferida u opcional según se expone en las reivindicaciones dependientes 2 a 13.

En otro ejemplo, la presente descripción se dirige a un dispositivo para la codificación de datos de vídeo como se define en las reivindicaciones 14 o 15.

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

BREVE DESCRIPCIÓN DE LOS DIBUJOS

La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y decodificación de vídeo de ejemplo que puede configurarse para realizar las técnicas de esta descripción.

La FIG. 2 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo que puede configurarse para realizar las técnicas de esta descripción.

La FIG. 3 es un diagrama de bloques que ilustra un ejemplo de decodificador de vídeo que puede configurarse para realizar las técnicas de esta descripción.

Las FIGS. 4A y 4B son diagramas conceptuales que ilustran un ejemplo de división de CTU a CU en HEVC, y la representación en árbol cuaternario correspondiente de la división de CTU a CU en HEVC.

La FIG. 5 es un diagrama conceptual que ilustra modos de división para unidades de codificación (CU) codificadas con modo interpredicción.

Las FIGS. 6A y 6B son diagramas conceptuales que ilustran aspectos de la estructura de división QTBT.

Las FIGS. 7A y 7B son diagramas conceptuales que ilustran un caso de uso de ejemplo de la estructura de división de bloques de árboles de tipos múltiples.

La FIG. 8 es un diagrama conceptual que ilustra ejemplos de palabras de código, expresadas en formato binario, que un dispositivo de codificación de vídeo puede señalar para los diversos esquemas de división que son posibles en la parte de PT de divisiones de bloques divididos en árboles de tipos múltiples.

La FIG. 9 es un diagrama conceptual que ilustra posiciones candidatas de varias ubicaciones vecinas superiores y ubicaciones vecinas izquierdas para modelización de contexto de un indicador de división de PT, según los aspectos de la presente descripción.

La FIG. 10 es un organigrama que ilustra un procedimiento de ejemplo que un dispositivo de codificación de vídeo puede realizar según distintos aspectos de la presente descripción.

La FIG. 11 es un organigrama que ilustra un procedimiento de ejemplo que un dispositivo de decodificación de vídeo puede realizar según distintos aspectos de la presente descripción.

DESCRIPCIÓN DETALLADA

La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y decodificación de vídeo de ejemplo que puede configurarse para realizar las técnicas de esta descripción para la predicción de vectores de movimiento. Como se muestra en la FIG. 1, el sistema 10 incluye un dispositivo de origen 12 que proporciona datos de vídeo codificados para ser decodificados en un momento posterior por un dispositivo de destino 14. En particular, el dispositivo de origen 12 proporciona los datos de vídeo al dispositivo de destino 14 a través de un medio legible por

ordenador 16. El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender cualquiera de una amplia gama de dispositivos, incluidos ordenadores de sobremesa, ordenadores portátiles (es decir, portátiles), ordenadores de tableta, decodificadores, teléfonos tales como los denominados teléfonos "inteligentes", los denominados paneles "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores multimedia digitales, consolas de videojuegos, dispositivos de transmisión de vídeo o similares. En algunos casos, el dispositivo de origen 12 y el dispositivo de destino 14 pueden estar equipados para la comunicación inalámbrica.

El dispositivo de destino 14 puede recibir los datos de vídeo codificados para decodificar 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 mover los datos de vídeo codificados desde el dispositivo de origen 12 al dispositivo de destino 14. En un ejemplo, el medio legible por ordenador 16 puede comprender un medio de comunicación para permitir que el dispositivo de origen 12 transmita datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real. Los datos de vídeo codificados se pueden modular según una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitirse al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrico o por cable, tal como un espectro de radiofrecuencia (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 de origen 12 al dispositivo de destino 14.

En algunos ejemplos, los datos codificados pueden emitirse desde la interfaz de salida 22 a un dispositivo de almacenamiento. De manera similar, se puede acceder a los datos codificados desde el dispositivo de almacenamiento mediante una interfaz de entrada. El dispositivo de almacenamiento puede incluir cualquiera de una variedad de medios de almacenamiento de datos distribuidos o de acceso local, tales como un disco duro, discos Blu-ray, DVD, CD-ROM, memoria flash, memoria volátil o no volátil, o cualquier otro medio de almacenamiento digital adecuado 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 de origen 12. El dispositivo de destino 14 puede acceder a los datos de vídeo almacenados desde el dispositivo de almacenamiento mediante transmisión en flujo continuo 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 de destino 14. Los servidores de archivos ilustrativos incluyen un servidor web (por ejemplo, para un sitio web), un servidor FTP, dispositivos de almacenamiento conectados a la red (Network Attached Storage, NAS) o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de cualquier conexión de datos convencional, 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 en flujo continuo, una transmisión de descarga o una combinación de estas.

Las técnicas de esta descripción no se limitan necesariamente a aplicaciones o configuraciones inalámbricas. Las técnicas se pueden aplicar a la codificación de vídeo en apoyo de cualquiera de una variedad de aplicaciones multimedia, tales como retransmisiones televisivas en abierto, retransmisiones de televisión por cable, retransmisiones de televisión por satélite, retransmisiones de vídeo de transmisión por Internet, tales como transmisión dinámica adaptativa a través de HTTP (Dynamic Adaptive Streaming over HTTP, DASH), vídeo digital que se codifica 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 se puede configurar para admitir la transmisión de vídeo unidireccional o bidireccional para admitir aplicaciones tales como transmisión de vídeo, reproducción de vídeo, difusión de vídeo y/o videotelefonía.

En el ejemplo de la FIG. 1, el dispositivo de origen 12 incluye la fuente de vídeo 18, el codificador de vídeo 20 y la interfaz de salida 22. El dispositivo de destino 14 incluye la interfaz de entrada 28, el decodificador de vídeo 30 y el dispositivo de visualización 32. Según esta descripción, el codificador de vídeo 20 del dispositivo de origen 12 puede configurarse para aplicar las técnicas de esta descripción para la predicción de vectores de movimiento. En otros ejemplos, un dispositivo de origen y un dispositivo de destino pueden incluir otros componentes o disposiciones. Por ejemplo, el dispositivo de origen 12 puede recibir datos de vídeo de una fuente de vídeo externa 18, tal como una cámara externa. Del mismo modo, el dispositivo de destino 14 puede interactuar con un dispositivo de visualización externo, en lugar de incluir un dispositivo de visualización integrado.

El sistema 10 ilustrado de la FIG. 1 es solo un ejemplo. Las técnicas de esta descripción para la predicción de vectores de movimiento pueden ser realizadas por cualquier dispositivo de codificación y/o decodificación de vídeo digital. Aunque generalmente las técnicas de esta descripción son realizadas por un dispositivo de codificación de vídeo, las técnicas también pueden ser realizadas por un codificador-decodificador de vídeo, normalmente referido como "CÓDEC". Además, las técnicas de esta descripción también se pueden realizar mediante un preprocesador de vídeo. El dispositivo de origen 12 y el dispositivo de destino 14 son simplemente ejemplos de dichos dispositivos de codificación donde el dispositivo de origen 12 genera datos de vídeo codificados para su transmisión al dispositivo de

destino 14. En algunos ejemplos, los dispositivos 12, 14 pueden funcionar de una manera sustancialmente simétrica de manera que cada uno de los dispositivos 12, 14 incluya componentes de codificación y decodificación de vídeo. Por lo tanto, el sistema 10 puede admitir transmisión de vídeo unidireccional o bidireccional entre dispositivos de vídeo 12, 14, por ejemplo, para vídeo en tiempo real, reproducción de vídeo, transmisión de vídeo o videotelefonía.

La fuente de vídeo 18 del dispositivo de origen 12 puede incluir un dispositivo de captura de vídeo, tal como una cámara de vídeo, un archivo de vídeo que contiene vídeo capturado previamente y/o una interfaz de alimentación de vídeo para recibir vídeo de un proveedor de contenido de vídeo. Como alternativa adicional, la fuente de vídeo 18 puede generar datos basados en gráficos informáticos como el vídeo de origen, o una combinación de vídeo en vivo, vídeo archivado y vídeo generado por ordenador. En algunos casos, si el origen de vídeo 18 es una cámara de vídeo, el dispositivo de origen 12 y el dispositivo de destino 14 pueden formar los llamados teléfonos con cámara o videoteléfonos. Sin embargo, como se mencionó anteriormente, las técnicas descritas en esta descripción se pueden aplicar a la codificación de vídeo en general, y se pueden usar en aplicaciones inalámbricas y/o por cable. En cada caso, el vídeo capturado, precapturado o generado por ordenador puede codificarse mediante el codificador de vídeo 20. La información de vídeo codificada puede a continuación enviarse mediante la interfaz de salida 22 a un medio legible por ordenador 16.

El medio legible por ordenador 16 puede incluir medios transitorios, tales como una difusión inalámbrica o transmisión de red por cable, o medios de almacenamiento (es decir, medios de almacenamiento no transitorios), tales como un disco duro, unidad flash, disco compacto, disco de vídeo digital, disco Blu-ray u otros medios legibles por ordenador. En algunos ejemplos, un servidor de red (no se muestra) puede recibir datos de vídeo codificados desde el dispositivo de origen 12 y proporcionar los datos de vídeo codificados al dispositivo de destino 14, por ejemplo, a través de una transmisión de red. De manera similar, un dispositivo informático de una instalación de producción de medios, como una instalación de estampado de discos, puede recibir datos de vídeo codificados desde el dispositivo de origen 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 y en diversos ejemplos.

La interfaz de entrada 28 del dispositivo de destino 14 recibe información del medio legible por ordenador 16. La información del medio legible por ordenador 16 puede incluir información sintáctica definida por el codificador de vídeo 20, que también es utilizada por el decodificador de vídeo 30, que incluye elementos sintácticos que describen características y/o procesamiento de bloques y otras unidades codificadas, por ejemplo, GOP. El dispositivo de visualización 32 muestra los datos de vídeo decodificados a un usuario, y puede comprender cualquiera de una variedad de dispositivos de visualización tales como un tubo de rayos catódicos (Cathode Ray Tube, CRT), una pantalla de cristal líquido (Liquid Crystal Display, LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (Organic Light Emitting Diode, OLED) u otro tipo de dispositivo de visualización.

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar según una norma de codificación de vídeo, tal como la norma de codificación de vídeo de alta eficiencia (HEVC), extensiones de la norma HEVC o normas posteriores, tales como ITU-T H.266. Como alternativa, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar según otras normas de propiedad exclusiva o industriales, como la norma ITU-T H.264, también conocida como MPEG-4, Parte 10, codificación de vídeo avanzada (Advanced Video Coding, AVC), o extensiones de dichas normas. Sin embargo, las técnicas de la presente descripción no se limitan a ninguna norma de codificación particular. Otros ejemplos de normas de codificación de vídeo incluyen MPEG-2 e ITU-T H.263. Aunque no se muestra en la FIG. 1, en algunos aspectos, el codificador de vídeo 20 y el decodificador de vídeo 30 se pueden integrar cada uno con un codificador y decodificador de audio, y pueden incluir unidades MUX-DEMUX adecuadas, u otro hardware y software, para manejar la codificación de audio y vídeo en un flujo de datos común o flujos de datos separados. Si corresponde, las unidades MUX-DEMUX pueden cumplir con el protocolo multiplexor ITU H.223 u otros protocolos como el protocolo de datagramas de usuario (User Datagram Protocol, UDP).

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementarse cada uno en uno cualquiera de diversos circuitos de codificador adecuados, como uno o más microprocesadores, procesadores de señales digitales (Digital Signal Processors, DSP), circuitos integrados específicos de aplicaciones (Application Specific Integrated Circuits, ASIC), matrices de puertas programables de campo (Field Programmable Gate Arrays, FPGA), lógica discreta, circuitos lógicos, circuitos de procesamiento, como circuitos de procesamiento de función fija y/o circuitos de procesamiento programables, hardware, firmware, software, o cualesquiera combinaciones de los mismos. Cuando las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio legible por ordenador no transitorio adecuado y ejecutar las instrucciones en hardware utilizando uno o más procesadores para realizar las técnicas de esta descripción. Cada uno del codificador de vídeo 20 y el decodificador de vídeo 30 se puede incluir en uno o más codificadores o decodificadores, cualquiera de los cuales se puede integrar como parte de un codificador/decodificador (CÓDEC) combinado en un dispositivo respectivo.

Los estándares de codificación de vídeo incluyen 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 multivista (MVC). Se describe un borrador conjunto de MVC en "Advanced video coding for generic audiovisual services", Recomendación

UIT-T H.264, marzo de 2010.

Además, existe un estándar de codificación de vídeo desarrollado recientemente, denominado codificación de vídeo de alta eficiencia (HEVC), desarrollado por el equipo de colaboración conjunta sobre codificación de vídeo (JCT-VC) del grupo de expertos en codificación de vídeo (VCEG) de la UIT-T y el grupo de expertos en cine (MPEG) de la ISO/IEC. Un borrador reciente de HEVC está disponible en http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v34.zip. La norma HEVC también se presenta conjuntamente en la Recomendación UIT-T H.265 y la Norma Internacional ISO/IEC 23008-2, ambas tituladas "Codificación de vídeo de alta eficiencia", y ambas publicadas en octubre de 2014.

El JCT-VC desarrolló el estándar HEVC. Los esfuerzos de estandarización de HEVC se basan en un modelo evolutivo de un dispositivo de codificación de vídeo denominado modelo de prueba HEVC (HEVC Model, HM). El HM supone varias capacidades adicionales de los dispositivos de codificación de vídeo en relación con los dispositivos existentes según, por ejemplo, ITU-T H.264/AVC. Por ejemplo, mientras que H.264 proporciona nueve modos de codificación de intrapredicción, HEVC HM puede proporcionar hasta treinta y tres modos de codificación de intrapredicción.

En general, el modelo de trabajo del HM describe que una trama o imagen de vídeo se puede dividir en una secuencia de bloques de árbol o unidades de codificación más grandes (Largest Coding Units, LCU) que incluyen muestras de luminancia y crominancia. Los datos de sintaxis dentro de un flujo de bits pueden definir un tamaño para la LCU, que es una unidad de codificación más grande en términos del número de píxeles. Un fragmento incluye una serie de bloques de árbol consecutivos en orden de codificación. Una trama o imagen de vídeo se puede dividir en uno o más fragmentos. Cada bloque de árbol se puede dividir en unidades de codificación (CU) según 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 sub-CU, el nodo correspondiente a la CU incluye cuatro nodos hoja, cada uno de los cuales corresponde a una de las sub-CU.

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 sub-CU. Los elementos de sintaxis para una CU pueden definirse de forma recursiva y pueden depender de si la CU se divide en sub-CU. Si una CU no se divide más, se denomina CU hoja. En esta descripción, cuatro sub-CU de una CU hoja también se denominarán 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 sub-CU de 8x8 también se denominarán CU hoja, aunque la CU de 16x16 nunca se haya dividido.

Una CU tiene un propósito similar al de un macrobloque de la norma H.264, con la salvedad de que una CU no tiene una distinción de tamaño. Por ejemplo, un bloque de árbol puede dividirse en cuatro nodos hijos (también denominados sub-CU), y cada nodo hijo puede ser a su vez un nodo padre y dividirse en otros cuatro nodos hijos. 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 en 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. Por consiguiente, un flujo de bits también puede definir una unidad de codificación más pequeña (Smallest Coding Unit, SCU). Esta descripción puede usar el término "bloque" para referirse a cualquiera de una CU, PU o TU, en el contexto de HEVC, o estructuras de datos similares en el contexto de otras normas (por ejemplo, macrobloques y sub-bloques de los mismos en H.264/AVC).

Una CU incluye un nodo de codificación y unidades de predicción (Prediction Units, PU) y unidades de transformación (Transform Units, TU) asociadas con el nodo de codificación. El tamaño de la CU corresponde al tamaño del nodo de codificación y debe tener forma cuadrada. El tamaño de la CU puede variar desde 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 con una CU pueden describir, por ejemplo, la división de la CU en una o más PU. Los modos de división pueden diferir entre si la CU está codificada en modo de salto o en modo directo, codificada en modo de intrapredicción o codificada en modo de interpretación. Las PU se pueden dividir para que no tengan forma cuadrada. Los datos de sintaxis asociados con una CU también pueden describir, por ejemplo, la división de la CU en una o más TU según un árbol cuaternario. Según la norma HEVC, una TU siempre tiene forma cuadrada. Es decir, cuando se aplica la transformada, se aplica una transformada unidimensional del mismo tamaño tanto horizontal como verticalmente.

La norma HEVC permite transformaciones según las TU, que pueden ser diferentes para diferentes CU. Las TU suelen tener un tamaño basado en el tamaño de las PU dentro de una CU determinada definida para una LCU dividida, aunque este puede no ser siempre el caso. Las TU suelen ser 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 conocida como "árbol cuaternario residual" (Residual Quad Tree, RQT). Los nodos hoja del RQT pueden denominarse unidades de transformación (TU). Los valores de diferencia de píxeles asociados con las TU se pueden transformar para producir coeficientes de transformación, que se pueden cuantificar.

Una CU hoja puede incluir una o más unidades de predicción (PU). En general, una PU representa un área espacial que corresponde a la totalidad o a una parte 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 tiene codificación intramodal, sus datos pueden incluirse en un árbol cuaternario residual (RQT), que puede incluir datos que describen un modo de intrapredicción para una TU correspondiente a la PU. Como otro ejemplo, cuando la PU tiene codificación intermodal, 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 (p. ej., precisión de un cuarto de píxel o 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 (p. ej., Lista 0 o Lista 1) para el vector de movimiento.

Una CU hoja que tiene una o más PU también puede incluir una o más unidades de transformación (TU). Las unidades de transformación pueden especificarse utilizando un RQT (también denominado estructura de árbol cuaternario de TU), como se ha descrito anteriormente. Por ejemplo, un indicador de división puede indicar si una CU hoja se divide en cuatro unidades de transformación. Luego, cada unidad de transformación puede dividirse aún más en sub-TU adicionales. Cuando una TU no se divide más, puede referirse como TU hoja. Generalmente, para la intracodificación, todas las TU hoja que pertenecen a una CU hoja comparten el mismo modo de intrapredicción. Es decir, generalmente se aplica el mismo modo de intrapredicción para calcular los valores predichos para todas las TU de una CU hoja. Para la intracodificación, un codificador de vídeo puede calcular un valor residual para cada TU hoja utilizando el modo de intrapredicción, como una diferencia entre la parte de la CU correspondiente a la TU y el bloque original. Una TU no se limita necesariamente al tamaño de una PU. Por lo tanto, las TU pueden ser más grandes o pequeñas que una PU. Para la intracodificación, una PU puede colocarse 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.

Además, las TU de las CU hoja también pueden estar asociadas con las respectivas estructuras de datos de árboles cuaternarios, denominadas árboles cuaternarios residuales (RQT). Es decir, una CU hoja puede incluir un árbol cuaternario que indique cómo se divide la CU hoja en TU. El nodo raíz de un árbol cuaternario 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). Las TU del RQT que no están divididas se denominan TU hoja. En general, esta descripción utiliza los términos CU y TU para referirse a CU hoja y TU hoja, respectivamente, a menos que se indique lo contrario.

Una secuencia de vídeo normalmente incluye una serie de tramas o imágenes de vídeo. Un grupo de imágenes (Group of Pictures, GOP) generalmente comprende una serie de una o más de las imágenes de vídeo. Un GOP puede incluir datos de sintaxis en un encabezado del GOP, un encabezado de una o más de las imágenes, o en otro lugar, que describe una serie de imágenes incluidas en el GOP. Cada fragmento de una imagen puede incluir datos de sintaxis de fragmento que describen un modo de codificación para el fragmento respectivo. El codificador de vídeo normalmente funciona en bloques de vídeo dentro de segmentos de vídeo individuales con el fin de codificar los datos de vídeo. Un bloque de vídeo puede corresponder a un nodo de codificación dentro de una CU. Los bloques de vídeo pueden tener tamaños fijos o variables, y pueden diferir en tamaño según una norma de codificación especificada.

Como ejemplo, el HM admite la predicción en varios tamaños de PU. Suponiendo que el tamaño de una CU particular es $2N \times 2N$, el HM admite intrapredicción en tamaños de PU de $2N \times 2N$ o $N \times N$, e interpredicción en tamaños de PU simétricos de $2N \times 2N$, $2N \times N$, $N \times 2N$, o $N \times N$. El HM también admite la división asimétrica para la interpredicción en tamaños de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$ y $nR \times 2N$. En la división asimétrica, una dirección de una CU no se divide, mientras que la otra dirección se divide en 25% y 75%. La parte de la CU correspondiente a la división del 25% se indica con una "n" seguida de una indicación de "Arriba", "Abajo", "Izquierda" o "Derecha". Por lo tanto, por ejemplo, " $2N \times nU$ " se refiere a una CU de $2N \times 2N$ que se divide horizontalmente con una PU de $2N \times 0,5N$ en la parte superior y una PU de $2N \times 1,5N$ en la parte inferior.

En esta descripción, " $N \times N$ " y " N por N " pueden usarse indistintamente para referirse a las dimensiones en píxeles de un bloque de vídeo en términos de dimensiones verticales y horizontales, por ejemplo, 16×16 píxeles o 16 por 16 píxeles. En general, un bloque de 16×16 tendrá 16 píxeles en dirección vertical ($y = 16$) y 16 píxeles en dirección horizontal ($x = 16$). Del mismo modo, un bloque $N \times N$ generalmente tiene N píxeles en una dirección vertical y N píxeles en una dirección horizontal, donde N representa un valor entero no negativo. Los píxeles de un bloque pueden disponerse en filas y columnas. Además, los bloques no necesitan tener necesariamente el mismo número de píxeles en la dirección horizontal que en la dirección vertical. Por ejemplo, los bloques pueden estar compuestos por $N \times M$ píxeles, donde M no es necesariamente igual a N .

Después de la codificación intrapredictiva o interpredictiva usando las PU de una CU, el codificador de vídeo puede calcular datos residuales para las TU de la CU. Las PU pueden comprender datos de sintaxis que describen una técnica o modo de generación de datos de píxeles predictivos en el dominio espacial (también denominado dominio de píxeles) y las TU pueden comprender coeficientes en el dominio de transformada después de la aplicación de una transformada, por ejemplo, una Transformada de Coseno Discreta (Discrete Cosine Transform, DCT), una

transformada entera, una transformada de ondícula o una transformada conceptualmente similar a datos de vídeo residuales. Los datos residuales pueden corresponder a diferencias de píxeles entre píxeles de la imagen no codificada y valores de predicción correspondientes a las PU. El codificador de vídeo 20 puede generar las unidades de transformación (TU), incluyendo los datos residuales de la unidad de control (CU), y luego transformarlas para generar coeficientes de transformación para la CU.

Tras cualquier transformación para generar coeficientes de transformación, el codificador de vídeo 20 puede cuantificarlos. La cuantificación generalmente se refiere a un proceso donde los coeficientes de transformación se cuantifican para posiblemente reducir la cantidad de datos utilizados al objeto de representar los coeficientes, proporcionando una compresión adicional. El procedimiento de cuantificación puede reducir la profundidad de bits asociada con algunos o todos los coeficientes. Por ejemplo, un valor de n bits puede redondearse a un valor de m bits durante la cuantificación, donde n es mayor que m .

Tras la cuantificación, el codificador de vídeo puede realizar un barrido de los coeficientes de transformación, generando un vector unidimensional a partir de la matriz bidimensional que incluye los coeficientes de transformación cuantificados. El barrido puede diseñarse para colocar coeficientes de energía más altos (y, por lo tanto, de frecuencia más baja) en la parte frontal de la matriz y para colocar coeficientes de energía más bajos (y, por lo tanto, de frecuencia más alta) en la parte posterior de la matriz. En algunos ejemplos, el codificador de vídeo 20 puede utilizar un orden de barrido predefinido para realizar un barrido de los coeficientes de transformada cuantificados con el fin de producir un vector serializado que puede estar sometido a codificación de entropía. En otros ejemplos, el codificador de vídeo 20 puede realizar un barrido adaptativo. Después de realizar un barrido de los coeficientes de transformación cuantificados para formar un vector unidimensional, el codificador de vídeo 20 puede codificar por entropía el vector unidimensional, por ejemplo, según la codificación de longitud variable adaptativa al contexto (Context-Adaptive Variable Length Coding, CAVLC), la codificación aritmética binaria adaptativa al contexto (Context-Adaptive Binary Arithmetic Coding, CABAC), la codificación aritmética binaria adaptativa al contexto basada en sintaxis (Syntax-based context-adaptive Binary Arithmetic Coding, SBAC), la codificación de entropía de división de intervalo de probabilidad (Probability Interval Partitioning Entropy, PIPE) u otra metodología de codificación de entropía. El codificador de vídeo 20 también puede codificar por entropía elementos sintácticos asociados con los datos de vídeo codificados para su uso por el decodificador de vídeo 30 en la decodificación de los datos de vídeo.

Para realizar CABAC, el codificador de vídeo 20 puede asignar un contexto dentro de un modelo de contexto a un símbolo para transmitir. El contexto puede referirse, por ejemplo, a si los valores vecinos del símbolo son distintos de cero o no. Para realizar CAVLC, el codificador de vídeo 20 puede seleccionar un código de longitud variable para un símbolo para transmitir. Las palabras de código en VLC pueden construirse de manera que los códigos relativamente más cortos correspondan a símbolos más probables, mientras que los códigos más largos correspondan a símbolos menos probables. De este modo, el uso de VLC puede conseguir un ahorro de bits, por ejemplo, usando palabras de código de igual longitud para cada símbolo que se transmitirá. La determinación de probabilidad puede basarse en un contexto asignado al símbolo. En algunos ejemplos, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden usar modelos afines en estimación y compensación de movimiento.

La FIG. 2 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo 20 que puede configurarse para realizar las técnicas de esta descripción para la predicción de vectores de movimiento. El codificador de vídeo 20 puede realizar una intracodificación e intercodificación de bloques de vídeo dentro de fragmentos de vídeo. La intracodificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial en el vídeo dentro de una trama o imagen de vídeo dada. La intercodificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal en vídeo dentro de tramas o imágenes adyacentes de una secuencia de vídeo. El intramodo (modo I) se puede referir a cualquiera de varios modos de codificación basados en el espacio. Los intermodos, como la predicción unidireccional (modo P) o la bipredicción (modo B), pueden referirse a cualquiera de varios modos de codificación basados en el tiempo.

Como se muestra en la FIG. 2, el codificador de vídeo 20 recibe un bloque de vídeo actual dentro de una trama para codificar. En el ejemplo de la FIG. 2, el codificador de vídeo 20 incluye una unidad de selección de modo 40, una memoria de imagen de referencia 64, un sumador 50, una unidad de procesamiento de transformación 52, una unidad de cuantificación 54 y una unidad de codificación de entropía 56. La unidad de selección de modo 40, a su vez, incluye la unidad de compensación de movimiento 44, la unidad de estimación de movimiento 42, la unidad de intrapredicción 46 y la unidad de división 48. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 también incluye la unidad de cuantificación inversa 58, la unidad de transformación inversa 60 y el sumador 62. En algunos ejemplos, también se incluye un filtro de eliminación de bloques (no mostrado en la Figura 2) para filtrar los límites de bloque para eliminar los artefactos de bloques del vídeo reconstruido. Si se desea, el filtro de eliminación de bloques filtraría típicamente la salida del sumador 62. También se pueden usar filtros adicionales (en bucle o posbucle) además del filtro de eliminación de bloques. Dichos filtros no se muestran en aras de la brevedad, pero si se desea, pueden filtrar la salida del sumador 50 (como un filtro en bucle).

Durante el proceso de codificación, el codificador de vídeo 20 recibe una trama o fragmento de vídeo que se va a codificar. La trama o fragmento se puede dividir en múltiples bloques de vídeo. La unidad de estimación de movimiento

42 y la unidad de compensación de movimiento 44 realizan codificación interpredictiva del bloque de vídeo recibido con respecto a uno o más bloques en una o más tramas de referencia para proporcionar predicción temporal. La unidad de intrapredicción 46 puede, alternativamente, realizar codificación intrapredictiva del bloque de vídeo recibido con respecto a uno o más bloques vecinos en el mismo fragmento o trama que el bloque que se va a codificar para proporcionar predicción espacial. El codificador de vídeo 20 puede realizar múltiples pasadas de codificación, por ejemplo, para seleccionar un modo de codificación apropiado para cada bloque de datos de vídeo.

Además, la unidad de división 48 puede dividir bloques de datos de vídeo en sub-bloques, basándose en la evaluación de esquemas de división anteriores en pases de codificación anteriores. Por ejemplo, la unidad de división 48 puede dividir inicialmente una trama o fragmento en LCU, y dividir cada una de las LCU en sub-CU basándose en el análisis de tasa-distorsión (por ejemplo, optimización de tasa-distorsión). La unidad de selección de modo 40 puede producir además una estructura de datos de árbol cuaternario indicativa de la división de una LCU en sub-CU. Las CU de nodo hoja del árbol cuaternario pueden incluir una o más PU y una o más TU.

La unidad de selección de modo 40 puede seleccionar uno de los modos de predicción, intra o inter, por ejemplo, basándose en los resultados de error, y proporciona el bloque predicho resultante al sumador 50 para generar datos residuales y al sumador 62 para reconstruir el bloque codificado para su uso como una imagen de referencia. La unidad de selección de modo 40 también proporciona elementos sintácticos, tales como vectores de movimiento, indicadores intramodo, información de división y otra información sintáctica similar, a la unidad de codificación de entropía 56.

La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden estar altamente integradas, pero se ilustran por separado con fines conceptuales. La estimación de movimiento, realizada por la unidad de estimación de movimiento 42, es el procedimiento de generación de vectores de movimiento, que estiman el movimiento para bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de una PU de un bloque de vídeo dentro de una trama o imagen de vídeo actual con respecto a un bloque predictivo dentro de una trama de referencia (u otra unidad codificada) en relación con el bloque actual que se codifica dentro de la trama actual (u otra unidad codificada). Un bloque predictivo es un bloque que coincide en gran medida con el bloque que se va a codificar, en términos de diferencia de píxeles, que puede determinarse mediante la suma de la diferencia absoluta (Sum of Absolute Difference, SAD), la suma de la diferencia cuadrada (Sum of Square Difference, SSD) u otras métricas de diferencia. En algunos ejemplos, el codificador de vídeo 20 puede calcular valores para posiciones de píxeles subenteros de imágenes de referencia almacenadas en la memoria de imágenes de referencia 64. Por ejemplo, el codificador de vídeo 20 puede interpolar valores de posiciones de un cuarto de píxel, posiciones de un octavo de píxel u otras posiciones de píxel fraccionarias de la imagen de referencia. Por lo tanto, la unidad de estimación de movimiento 42 puede realizar una búsqueda de movimiento con respecto a las posiciones de píxel completas y las posiciones de píxel fraccionarias y emitir un vector de movimiento con precisión de píxel fraccionario.

La unidad de estimación de movimiento 42 calcula un vector de movimiento para una PU de un bloque de vídeo en un fragmento intercodificado comparando la posición de la PU con la posición de un bloque predictivo de una imagen de referencia. La imagen de referencia puede seleccionarse de una primera lista de imágenes de referencia (lista 0) o una segunda lista de imágenes de referencia (lista 1), cada una de las cuales identifica una o más imágenes de referencia almacenadas en la memoria de imágenes de referencia 64. La unidad de estimación de movimiento 42 envía el vector de movimiento calculado a la unidad de codificación de entropía 56 y la unidad de compensación de movimiento 44.

La compensación de movimiento, realizada por la unidad de compensación de movimiento 44, puede implicar buscar o generar el bloque predictivo basándose en el vector de movimiento determinado por la unidad de estimación de movimiento 42. De nuevo, la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden integrarse funcionalmente, en algunos ejemplos. Al recibir el vector de movimiento para el PU del bloque de vídeo actual, la unidad de compensación de movimiento 44 puede ubicar el bloque predictivo al que apunta el vector de movimiento en una de las listas de imágenes de referencia. El sumador 50 forma un bloque de vídeo residual restando los valores de píxeles del bloque predictivo de los valores de píxeles del bloque de vídeo actual que se está codificando, formando valores de diferencia de píxeles, como se analiza a continuación. En general, la unidad de estimación de movimiento 42 realiza la estimación de movimiento con respecto a los componentes de luminancia, y la unidad de compensación de movimiento 44 utiliza vectores de movimiento calculados basándose en los componentes de luminancia tanto para los componentes de crominancia como para los componentes de luminancia. La unidad de selección de modo 40 también puede generar elementos sintácticos asociados con los bloques de vídeo y el fragmento de vídeo para su uso por el decodificador de vídeo 30 en la decodificación de los bloques de vídeo del fragmento de vídeo.

El codificador de vídeo 20 se puede configurar para realizar cualquiera de las diversas técnicas de esta descripción descritas anteriormente con respecto a la FIG. 1, y como se describirá con más detalle a continuación. Por ejemplo, la unidad de compensación de movimiento 44 puede configurarse para codificar información de movimiento para un bloque de datos de vídeo usando AMVP o modo de fusión según las técnicas de la presente descripción.

Suponiendo que la unidad de compensación de movimiento 44 elige realizar el modo de fusión, la unidad de compensación de movimiento 44 puede formar una lista de candidatos que incluye un conjunto de candidatos de fusión. La unidad de compensación de movimiento 44 puede añadir candidatos a la lista de candidatos basándose en un orden predeterminado particular. La unidad de compensación de movimiento 44 también puede añadir candidatos adicionales y realizar la poda de la lista de candidatos, como se ha analizado anteriormente. En última instancia, la unidad de selección de modo 40 puede determinar cuál de los candidatos se utilizará para codificar información de movimiento del bloque actual, y codificar un índice de fusión que representa el candidato seleccionado.

La unidad de intrapredicción 46 puede realizar intrapredicción de un bloque actual, como una alternativa a la interpredicción realizada por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44, como se ha descrito anteriormente. En particular, la unidad de intrapredicción 46 puede determinar un modo de intrapredicción para su uso para codificar un bloque actual. En algunos ejemplos, la unidad de intrapredicción 46 puede codificar un bloque actual utilizando varios modos de intrapredicción, por ejemplo, durante pases de codificación separados, y la unidad de intrapredicción 46 (o unidad de selección de modo 40, en algunos ejemplos) puede seleccionar un modo de intrapredicción apropiado para su uso entre los modos probados.

Por ejemplo, la unidad de intrapredicción 46 puede calcular valores de tasa-distorsión usando un análisis de tasa-distorsión para los diversos modos de intrapredicción probados, y seleccionar el modo de intrapredicción que tiene las mejores características de tasa-distorsión entre los modos probados. El análisis de tasa-distorsión determina generalmente una cantidad de distorsión (o error) entre un bloque codificado y un bloque original no codificado que fue codificado para producir el bloque codificado, al igual que como tasa de bits (es decir, el número de bits) usados para producir el bloque codificado. La unidad de intrapredicción 46 puede calcular relaciones a partir de las distorsiones y tasas para los diversos bloques codificados al objeto de determinar qué modo de intrapredicción muestra el mejor valor de tasa-distorsión para el bloque.

Después de seleccionar un modo de intrapredicción para un bloque, la unidad de intrapredicción 46 puede proporcionar información indicativa del modo de intrapredicción seleccionado para el bloque a la unidad de codificación de entropía 56. La unidad de codificación de entropía 56 puede codificar la información que indica el modo de intrapredicción seleccionado. El codificador de vídeo 20 puede incluir en los datos de configuración de flujo de bits transmitidos, que pueden incluir una pluralidad de tablas de índices de modo de intrapredicción y una pluralidad de tablas de índices de modo de intrapredicción modificadas (también denominadas tablas de mapeo de palabras de código), definiciones de contextos de codificación para diversos bloques e indicaciones de un modo de intrapredicción más probable, una tabla de índices de modo de intrapredicción y una tabla de índices de modo de intrapredicción modificada para usar para cada uno de los contextos.

El codificador de vídeo 20 forma un bloque de vídeo residual restando los datos de predicción de la unidad de selección de modo 40 del bloque de vídeo original que se está codificando. El sumador 50 representa el componente o componentes que realizan esta operación de resta. La unidad de procesamiento de transformación 52 aplica una transformación, tal como una transformación de coseno discreta (DCT) o una transformación conceptualmente similar, al bloque residual, produciendo un bloque de vídeo comprendiendo valores de coeficientes de transformación. La unidad de procesamiento de transformación 52 puede realizar otras transformaciones que son conceptualmente similares a DCT. Se podrían utilizar transformaciones de ondículas, transformaciones de enteros, transformaciones de sub-bandas, DST u otros tipos de transformaciones en lugar de una DCT.

En cualquier caso, la unidad de procesamiento de transformación 52 aplica la transformación al bloque residual, produciendo un bloque de coeficientes de transformación residuales. La transformación puede convertir la información residual de un dominio de píxeles a un dominio de transformación, tal como un dominio de frecuencia. La unidad de procesamiento de transformación 52 puede enviar los coeficientes de transformación resultantes a la unidad de cuantificación 54. La unidad de cuantificación 54 cuantifica los coeficientes de transformada para reducir aún más la tasa de bits. El procedimiento de cuantificación puede reducir la profundidad de bits asociada con algunos o todos los coeficientes. El grado de cuantificación puede modificarse ajustando un parámetro de cuantificación. En algunos ejemplos, la unidad de cuantificación 54 puede realizar un barrido de la matriz que incluye los coeficientes de transformación cuantificados. De manera alternativa, la unidad de codificación de entropía 56 puede realizar el barrido.

Después de la cuantificación, la unidad de codificación de entropía 56 codifica por entropía los coeficientes de transformación cuantificados. Por ejemplo, la unidad de codificación de entropía 56 puede realizar codificación de longitud variable adaptativa al contexto (CAVLC), codificación aritmética binaria adaptativa al contexto (CABAC), codificación aritmética binaria adaptativa al contexto basada en sintaxis (SBAC), codificación de entropía de división de intervalo de probabilidad (PIPE) u otra técnica de codificación de entropía. En el caso de la codificación de entropía basada en el contexto, el contexto puede basarse en bloques vecinos. Después de la codificación de entropía mediante la unidad de codificación de entropía 56, el flujo de bits codificado puede transmitirse a otro dispositivo (por ejemplo, el decodificador de vídeo 30) o archivar para su posterior transmisión o recuperación.

La unidad de cuantificación inversa 58 y la unidad de transformación inversa 60 aplican cuantificación y transformación inversas, respectivamente, para reconstruir el bloque residual en el dominio de píxeles, por ejemplo, para su uso

posterior como bloque de referencia. La unidad de compensación de movimiento 44 puede calcular un bloque de referencia añadiendo el bloque residual a un bloque predictivo de una de las tramas de la memoria de imágenes de referencia 64. La unidad de compensación de movimiento 44 también puede aplicar uno o más filtros de interpolación al bloque residual reconstruido para calcular valores de píxeles subenteros para su uso en la estimación de movimiento. El sumador 62 añade el bloque residual reconstruido al bloque de predicción con compensación de movimiento producido anteriormente por la unidad de compensación de movimiento 44 para producir un bloque de vídeo reconstruido para su almacenamiento en la memoria de imágenes de referencia 64. El bloque de vídeo reconstruido puede ser utilizado por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 como un bloque de referencia para intercodificar un bloque en una trama de vídeo posterior.

La FIG. 3 es un diagrama de bloques que ilustra un ejemplo de decodificador de vídeo 30 que puede configurarse para realizar las técnicas de predicción de vector de movimiento de esta descripción. En el ejemplo de la FIG. 3, el decodificador de vídeo 30 incluye una unidad de decodificación entrópica 70, una unidad de compensación de movimiento 72, una unidad de intrapredicción 74, una unidad de cuantificación inversa 76, una unidad de transformación inversa 78, una memoria de imágenes de referencia 82 y un sumador 80. El decodificador de vídeo 30 puede, en algunos ejemplos, realizar una pasada de decodificación generalmente recíproca a la pasada de codificación descrita con respecto al codificador de vídeo 20 (FIG. 2). La unidad de compensación de movimiento 72 puede generar datos de predicción basados en vectores de movimiento recibidos de la unidad de decodificación entrópica 70, mientras que la unidad de intrapredicción 74 puede generar datos de predicción basados en indicadores de modo de intrapredicción recibidos de la unidad de decodificación entrópica 70.

Durante el proceso de decodificación, el decodificador de vídeo 30 recibe un flujo de bits de vídeo codificado que representa bloques de vídeo de un fragmento de vídeo codificado y elementos sintácticos asociados del codificador de vídeo 20. La unidad de decodificación entrópica 70 del decodificador de vídeo 30 decodifica por entropía el flujo de bits para generar coeficientes cuantificados, vectores de movimiento o indicadores de modo de intrapredicción y otros elementos sintácticos. La unidad de decodificación entrópica 70 reenvía los vectores de movimiento y otros elementos sintácticos a la unidad de compensación de movimiento 72. El decodificador de vídeo 30 puede recibir los elementos sintácticos a nivel de fragmento de vídeo y/o a nivel de bloque de vídeo.

Cuando el fragmento de vídeo se codifica como un fragmento intracodificado (I), la unidad de intrapredicción 74 puede generar datos de predicción para un bloque de vídeo del fragmento actual, basándose en un modo de intrapredicción señalado y en datos de bloques previamente decodificados de la trama o imagen actual. Cuando la trama de vídeo se codifica como un fragmento intercodificado (es decir, B, P o GPB), la unidad de compensación de movimiento 72 genera bloques predictivos para un bloque de vídeo del fragmento actual, basándose en los vectores de movimiento y otros elementos sintácticos recibidos de la unidad de decodificación de entropía 70. Los bloques predictivos pueden producirse a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El decodificador de vídeo 30 puede construir las listas de tramas de referencia, la lista 0 y la lista 1, utilizando técnicas de construcción predeterminadas basadas en imágenes de referencia almacenadas en la memoria de imágenes de referencia 82.

La unidad de compensación de movimiento 72 determina la información de predicción para un bloque de vídeo del fragmento de vídeo actual mediante el análisis de los vectores de movimiento y otros elementos sintácticos, y utiliza la información de predicción para producir los bloques predictivos para el bloque de vídeo actual que se está decodificando. Por ejemplo, la unidad de compensación de movimiento 72 usa parte de los elementos de sintaxis recibidos para determinar un modo de predicción (p. ej., intra- o inter-predicción) utilizado para codificar los bloques de vídeo del fragmento de vídeo, un tipo de fragmento de interpredicción (p. ej., fragmento B, fragmento P), información de construcción para una o más de las listas de imágenes de referencia para el fragmento, vectores de movimiento para cada bloque de vídeo intercodificado del fragmento, estado de interpredicción para cada bloque de vídeo intercodificado del fragmento y otra información para decodificar los bloques de vídeo en el fragmento de vídeo actual.

La unidad de compensación de movimiento 72 también puede realizar una interpolación basada en filtros de interpolación. La unidad de compensación de movimiento 72 puede utilizar filtros de interpolación como los utilizados por el codificador de vídeo 20 durante la codificación de los bloques de vídeo para calcular valores interpolados para píxeles subenteros de bloques de referencia. En este caso, la unidad de compensación de movimiento 72 puede determinar los filtros de interpolación utilizados por el codificador de vídeo 20 a partir de los elementos sintácticos recibidos y utilizar los filtros de interpolación para producir bloques predictivos.

El decodificador de vídeo 30 se puede configurar para realizar cualquiera de las diversas técnicas de esta descripción expuestas anteriormente con respecto a la FIG. 1, y como se tratará con más detalle a continuación. Por ejemplo, la unidad de compensación de movimiento 72 se puede configurar para determinar realizar predicción de vector de movimiento usando AMVP o modo de fusión según las técnicas de esta descripción. La unidad de decodificación entrópica 70 puede decodificar uno o más elementos sintácticos que representan cómo se codifica la información de movimiento para el bloque actual.

Suponiendo que los elementos sintácticos indican que se realiza el modo de fusión, la unidad de compensación de

movimiento 72 puede formar una lista de candidatos que incluye un conjunto de candidatos de fusión. La unidad de compensación de movimiento 72 puede añadir candidatos a la lista de candidatos basándose en un orden predeterminado particular. La unidad de compensación de movimiento 72 también puede añadir candidatos adicionales y realizar la poda de la lista de candidatos, como se ha analizado anteriormente. En última instancia, la unidad de compensación de movimiento 72 puede decodificar un índice de fusión que representa cuál de los candidatos se utiliza para codificar la información de movimiento para el bloque actual.

La unidad de cuantificación inversa 76 cuantifica inversamente, es decir, descuantifica, los coeficientes de transformación cuantificados proporcionados en el flujo de bits y decodificados por entropía por la unidad de decodificación de entropía 70. El proceso de cuantificación inversa puede incluir el uso de un parámetro de cuantificación QP_Y calculado por el decodificador de vídeo 30 para cada bloque de vídeo en el fragmento de vídeo para determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa que debe aplicarse.

La unidad de transformación inversa 78 aplica una transformación inversa, por ejemplo, una DCT inversa, una transformación entera inversa o un proceso de transformación inversa conceptualmente similar, a los coeficientes de transformación para producir bloques residuales en el dominio de píxeles.

Después de que la unidad de compensación de movimiento 72 genera el bloque predictivo para el bloque de vídeo actual basándose en los vectores de movimiento y otros elementos sintácticos, el decodificador de vídeo 30 forma un bloque de vídeo decodificado sumando los bloques residuales de la unidad de procesamiento de transformada inversa 78 con los bloques predictivos correspondientes generados por la unidad de compensación de movimiento 72. El sumador 80 representa el componente o componentes que realizan esta operación de suma. Si se desea, también se puede aplicar un filtro de eliminación de bloques para filtrar los bloques decodificados con el fin de eliminar los artefactos de bloques. También se pueden usar otros filtros de bucle (ya sea en el bucle de codificación o después del bucle de codificación) para suavizar las transiciones de píxeles o mejorar la calidad del vídeo. Los bloques de vídeo decodificados en una trama o imagen dadas se almacenan luego en la memoria de imágenes de referencia 82, que almacena imágenes de referencia utilizadas para la compensación de movimiento posterior. La memoria de imágenes de referencia 82 también almacena vídeo decodificado para su presentación posterior en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la Figura 1.

En HEVC, la más grande unidad de codificación en un fragmento se denomina unidad de árbol de codificación (Coding Tree Unit, CTU). Una CTU contiene un árbol cuaternario. Los nodos del árbol cuaternario se refieren como unidades de codificación (CU). Además, una CTU contiene un bloque de árbol de codificación de luminancia (Coding Tree Block, CTB) y dos CTB de crominancia, así como los elementos de sintaxis asociados. El tamaño de un CTB de luminancia puede estar en un rango de 16x16 a 64x64 según el perfil principal de HEVC. Sin embargo, se observará que técnicamente también pueden admitirse tamaños de CTB de 8x8. Una CTU puede dividirse recursivamente en unidades de codificación (CU) en forma de un árbol cuaternario, como la estructura de árbol cuaternario mostrada en las FIGS. 4A y 4B, que se describen más adelante.

Las FIGS. 4A y 4B son diagramas conceptuales que ilustran un ejemplo de división de CTU a CU en HEVC, y la representación en árbol cuaternario correspondiente de la división de CTU a CU en HEVC. Es decir, la profundidad de división de cada rama y sub-rama de la FIG. 4B corresponde a la división de árbol cuaternario que produce las CU ilustradas en la FIG. 4A. Uno de los sub-bloques ilustrados en la FIG. 4A está sombreado, para ilustrar que el sub-bloque sombreado es un ejemplo de una CU de 8x8. Para el sub-bloque sombreado, no se permiten más divisiones, y con ello, no puede requerirse una señalización relacionada con divisiones. En los casos en que los nodos hoja ilustrados en la FIG. 4B (p. ej., las CU resultantes de la división en la FIG. 4A) corresponden a una CU de 8x8, no se requiere señalización. Se describen aspectos de la representación en árbol cuaternario en "Improved Video Compression Efficiency Through Flexible Unit Representation and Corresponding Extension of Coding Tools" de W. J. Han y col., IEEE Transaction on Circuits and Systems for Video Technology, vol. 20, nº 12, pp. 1709-1720, diciembre de 2010. La señalización de la división de CTU a CU de la FIG. 4A sigue la estructura de árbol cuaternario ilustrada en la FIG. 4B, donde cada nodo respectivo en el árbol cuaternario ilustrado consume un bit para indicar si el nodo respectivo debería o no dividirse adicionalmente. Por ejemplo, el codificador de vídeo 20 puede señalar el esquema de división de CTU a CU de la FIG. 4A al decodificador de vídeo 30 según las características de señalización descritas anteriormente.

El tamaño de CU podría ser el mismo que el tamaño de un CTB, aunque el tamaño de CU puede ser de apenas 8x8. Cada unidad de codificación (CU) se codifica según un modo. Por ejemplo, el codificador de vídeo 20 puede codificar, y el decodificador de vídeo 30 puede decodificar, cada CU según un tipo intramodo o intermodo. Si una CU se codifica en intermodo (es decir, el codificador de vídeo 20 aplica intermodo en la codificación de la CU), la CU puede dividirse adicionalmente en dos (2) o cuatro (4) unidades de predicción (PU), o puede convertirse en solo una (1) PU cuando no se aplica división adicional. Cuando hay dos PU presentes en una CU, las PU pueden ser rectángulos, cada uno de los cuales cubre la mitad del tamaño (área) de la CU, o las PU pueden ser dos rectángulos con tamaños respectivos (p. ej., áreas) con un cuarto (1/4) y tres cuartos partes (3/4) del tamaño de la CU.

La FIG. 5 es un diagrama conceptual que ilustra modos de división para un modo interpredicción. Existen ocho (8)

modos de división para CU codificadas con el modo interpredicción, como se muestra en la FIG. 5. Los ocho (8) modos de división mostrados en la FIG. 5 son PART_2Nx2N, PART_2NxN, PART_Nx2N, PART_NxN, PART_2NxN_U, PART_2NxN_D, PART_nLx2N y PART_nRx2N.

En los casos en que una CU en particular se codifica en intermodo, para cada PU está presente un conjunto de información de movimiento. Por ejemplo, si el codificador de vídeo 20 realiza intercodificación para la CU, entonces el codificador de vídeo 20 puede señalar un conjunto de información de movimiento al decodificador de vídeo 30 para la CU. Además, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden codificar (p. ej., codificar o decodificar, respectivamente) cada PU con un único modo de interpredicción para deducir el conjunto respectivo de información de movimiento. En los casos en que una CU se somete a intracodificación, 2Nx2N y NxN son las únicas formas de PU permisibles. En estos casos, dentro de cada PU, se codifica un único modo de intrapredicción, mientras que el codificador de vídeo 20 señala el modo de predicción de crominancia en el nivel de CU. Las formas de PU intra NxN están permitidas solo cuando el tamaño real de la CU actual es igual al tamaño de CU más pequeño en el conjunto de parámetros de secuencia (Sequence Parameter Set, SPS) correspondiente.

En el curso del desarrollo más allá de HEVC y otras normas existentes se han propuesto y estudiado diversos esquemas de división. Un ejemplo es la estructura de árbol cuaternario-árbol binario (Quad-Tree Binary-Tree), que se describe en mayor detalle más adelante. En la propuesta VCEG COM16-C966 ("Block partitioning structure for next generation video coding" de J. An, Y.-W. Chen, K. Zhang, H. Huang, Y.-W. Huang, y S. Lei, International Telecommunication Union, COM16-C966, septiembre de 2015), se propuso una estructura de árbol cuaternario-árbol binario (QTBT) para futuras normas de codificación de vídeo posteriores a HEVC. Las simulaciones mostraron que la estructura QTBT propuesta es más eficiente que la estructura de árbol cuaternario en la norma HEVC usada.

En la estructura QTBT propuesta de COM16-C966, primero se divide un CTB según la estructura de árbol cuaternario, donde la división de árbol cuaternario de un nodo puede iterarse hasta que el nodo alcanza un tamaño de nodo hoja de árbol cuaternario mínimo permitido (MinQTSIZE). Según la estructura QTBT, si el tamaño de nodo hoja de árbol cuaternario no es mayor que un tamaño de nodo raíz del árbol binario máximo permitido (MaxBTSIZE), puede dividirse adicionalmente según una estructura de árbol binario. La división de árbol binario de un nodo puede iterarse hasta que el nodo alcanza el tamaño de nodo hoja de árbol binario mínimo permitido (MinBTSIZE) o la profundidad de árbol binario máxima permitida (MaxBTDDEPTH). El nodo hoja de árbol binario es, en concreto, una CU que se puede usar para la predicción (por ejemplo, predicción intra-imagen o inter-imagen) y transformarse sin ninguna división adicional.

Según la división de árbol binario, existen dos tipos de división, en concreto, división horizontal simétrica y división vertical simétrica. En un ejemplo de la estructura de división QTBT, el tamaño de la CTU se establece en 128x128 (es decir, 128x128 muestras de luminancia y dos muestras de crominancia 64x64 correspondientes), el MinQTSIZE se establece en 16x16, el MaxBTSIZE se establece en 64x64, el MinBTSIZE (tanto para la anchura como para la altura) se establece en 4 y el MaxBTDDEPTH se establece en 4. La división de árbol cuaternario se aplica primero a la CTU, para generar nodos hoja de árbol cuaternario. Los nodos hoja de árbol cuaternario pueden tener un tamaño comprendido entre 16x16 (es decir, el MinQTSIZE) y 128x128 (es decir, el tamaño de la CTU). Si el nodo hoja de árbol cuaternario es 128x128, entonces el nodo hoja de árbol cuaternario no se dividirá más según la estructura de árbol binario, dado que el tamaño de nodo hoja de árbol cuaternario supera el MaxBTSIZE (es decir, 64x64).

En caso contrario (p. ej., si el tamaño de nodo hoja de árbol cuaternario no supera el MaxBTSIZE de 64x64), el nodo hoja de árbol cuaternario puede dividirse adicionalmente según la estructura de árbol binario. Por tanto, el nodo hoja de árbol cuaternario es también el nodo raíz para el árbol binario, y tiene una profundidad de árbol binario de 0. Cuando la profundidad de árbol binario alcanza MaxBTDDEPTH (es decir, 4) por medio de una división binaria iterativa, implica que no existe división adicional de ninguna clase. Si el nodo de árbol binario tiene una anchura igual a MinBTSIZE (es decir, 4), implica ausencia de división vertical adicional. De forma similar, si el nodo de árbol binario tiene una altura igual al MinBTSIZE (4), implica ausencia de división horizontal adicional. Los nodos hoja del árbol binario son, en concreto, CU procesadas adicionalmente por predicción y transformación sin ninguna división adicional.

Las FIGS. 6A y 6B son diagramas conceptuales que ilustran aspectos de la estructura de división QTBT. La FIG. 6A ilustra un ejemplo de división de bloque según la estructura QTBT. De este modo, la FIG. 6A puede describirse como una ilustración de la estructura QTBT. La FIG. 6B ilustra la estructura en árbol correspondiente a la división de bloque basada en QTBT mostrada en la FIG. 6A. En la FIG. 6B, las líneas continuas indican división basada en árbol cuaternario, y las líneas discontinuas indican división basada en árbol binario. En cada nodo de división (es decir, nodo no hoja) del árbol binario parte de la FIG. 6B, el codificador de vídeo 20 puede señalar un indicador para indicar el tipo de división binaria (es decir, horizontal o vertical) que se utiliza. En el ejemplo de la FIG. 6B, un valor de indicador de cero (0) indica división horizontal, y un valor de indicador de uno (1) indica división vertical. Para la parte basada en árbol cuaternario de la estructura QTBT, no es necesario señalar una indicación del tipo de división, ya que los nodos de división de la parte basada en árbol cuaternario se dividen siempre horizontal y verticalmente en cuatro (4) sub-bloques con tamaños iguales.

Otra estructura de árbol que se ha propuesto y estudiado posteriormente para HEVC se refiere como "estructura de árboles de tipos múltiples". En las solicitudes de patente provisionales de EE.UU. nº 62/279,233 y 62/311,248 se

propuso y se describió la estructura de árboles de tipos múltiples. Con las técnicas descritas en las solicitudes de patente provisionales de EE.UU. nº 62/279,233 y 62/311,248, un nodo de árbol puede dividirse adicionalmente usando múltiples tipos de árboles, como la estructura de árbol binario, árbol ternario simétrico del lado central y/o árbol cuaternario. Según la estructura de árboles de tipos múltiples de dos niveles, primero se construye un árbol de región (Region Tree, RT) con divisiones árbol cuaternario de una CTU. La parte de RT de la estructura de árboles de tipos múltiples se sigue de la construcción de una parte de árbol de predicción (PT) de la estructura de árboles de tipos múltiples. En la parte de PT de la estructura de árboles de tipos múltiples, solo el árbol binario y el árbol ternario simétrico del lado central pueden expandirse.

Es decir, en la parte de PT de la estructura de árboles de tipos múltiples, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden dividir los nodos de división según una entre la estructura de árbol binario o la estructura de árbol ternario del lado central. En la parte de PT de división basada en árboles de tipos múltiples, un nodo de división que se obtuvo como resultado de una división basada en árbol binario puede dividirse adicionalmente según la estructura de árbol ternario del lado central. Además, en la parte de PT de división basada en árboles de tipos múltiples, un nodo de división que se produjo como resultado de una división basada en árbol ternario del lado central puede dividirse adicionalmente según la estructura de árbol binario.

Las FIGS. 7A y 7B son diagramas conceptuales que ilustran un caso de uso de ejemplo de la estructura de división de bloques de árboles de tipos múltiples. La FIG. 7A ilustra un ejemplo de un bloque que se divide según la estructura de árboles de tipos múltiples. De este modo, la FIG. 7A puede describirse como una ilustración de la estructura de división de árboles de tipos múltiples. La FIG. 7B ilustra la estructura en árbol correspondiente a la división de bloques basada en árboles de tipos múltiples mostrada en la FIG. 7A.

En cada una de las FIGS. 7A y 7B, las líneas continuas se usan para ilustrar la división según los aspectos basados en RT de la estructura de división de árboles de tipos múltiples, mientras que las líneas discontinuas se usan para ilustrar la división según los aspectos basados en PT de la estructura de división de árboles de tipos múltiples. Como se muestra en las FIGS. 7A y 7B, según la estructura de división de árboles de tipos múltiples, el codificador de vídeo 20 y el decodificador de vídeo 30 empiezan la división basada en PT para una rama particular solo después de concluir la división basada en RT para esa rama particular. Además, como se ilustra en las FIGS. 7A y 7B, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementar tanto la división basada en árbol binario como división basada en árbol ternario del lado central múltiples veces, y en cualquier orden, en la parte basada en PT de un esquema de división basada en árboles de tipos múltiples.

Como se muestra en la FIG. 7A, el bloque 90 se divide primero según la estructura de árbol cuaternario, en cuatro sub-bloques cuadrados. La división de árbol cuaternario del bloque 90 se ilustra en la FIG. 7B por medio de una división en línea continua de cuatro ramas del nodo raíz. De izquierda a derecha, cada una de las cuatro ramas que se dividen desde el nodo raíz del árbol de división 92 (en la FIG. 7B) corresponden respectivamente a los sub-bloques superior izquierdo, superior derecho, inferior izquierdo e inferior derecho resultantes de la división de árbol cuaternario (es decir, primera parte de RT) del bloque 90. El sub-bloque superior izquierdo se divide de nuevo según la división de estructura de árbol cuaternario, y cada uno de los sub-bloques resultantes representa un nodo hoja (es decir, no se divide más). Como toda la división del sub-bloque superior izquierdo del bloque 90 se realiza según la división de estructura de árbol cuaternario, la rama más a la izquierda de árbol de división 92 concluye dentro de la parte de RT del esquema de división de árboles de tipos múltiples. La naturaleza limitada de RT de la rama más a la izquierda del árbol 92 se muestra en la FIG. 7B por medio de líneas continuas que se usan para ilustrar toda la división que afecta a la rama más a la izquierda del árbol 92.

El sub-bloque superior derecho del bloque 92 se divide según la estructura de división de árbol ternario del lado central, para formar tres sub-bloques rectangulares. Más en concreto, en el caso del sub-bloque superior derecho del bloque 92, el codificador de vídeo 20 y el decodificador de vídeo 30 implementan una división vertical según la estructura de división de árbol ternario del lado central. Es decir, el codificador de vídeo 20 y el decodificador de vídeo 30 dividen el sub-bloque superior derecho verticalmente, para formar los tres sub-bloques rectangulares resultantes. La división basada en el árbol ternario vertical del lado central del sub-bloque superior derecho se muestra en el árbol de división 92 como una división de tres ramas de la segunda rama desde la izquierda de la división del nodo raíz. Como el sub-bloque superior derecho del bloque 90 se divide según la estructura de división de árbol ternario del lado central, la división del sub-bloque superior derecho forma parte de la parte de PT del esquema basado en árbol de tipos múltiples del árbol de división 92. De este modo, la división basada en árbol ternario del lado central en la segunda rama desde la izquierda de árbol de división 92 se ilustra usando una línea discontinua en la FIG. 7B.

A su vez, el codificador de vídeo 20 y el decodificador de vídeo 30 implementan la división basada en árbol binario del sub-bloque rectangular más a la izquierda del sub-bloque superior derecho, para formar dos sub-bloques rectangulares. Más en concreto, el codificador de vídeo 20 y el decodificador de vídeo 30 implementan una versión horizontal de la división basada en árbol binario, con respecto al sub-bloque más a la izquierda del sub-bloque superior derecho del bloque 90. La división basada en árbol binario se denota en el árbol de división 92 usando una división bidireccional en línea discontinua del nodo más a la izquierda que se produjo como resultado de la división de la segunda rama desde la izquierda.

El sub-bloque inferior izquierdo del bloque 90 se divide horizontalmente según la estructura de división de árbol ternario del lado central, y el sub-bloque medio resultante de esta división se divide adicionalmente en sentido vertical según la estructura de división de árbol ternario del lado central. Estos aspectos de la parte de PT de la división de árboles de tipos múltiples del bloque 90 se muestran usando líneas discontinuas en la división descendente de la rama tercera desde la izquierda que se origina en el nodo raíz del árbol de división 92.

El sub-bloque inferior derecho del bloque 90 se divide horizontalmente según la estructura de división de árbol binario, y el sub-bloque izquierdo resultante de esta división se divide adicionalmente en sentido vertical según la estructura de división de árbol ternario del lado central. Estos aspectos de la parte de PT de la división de árboles de tipos múltiples del bloque 90 se muestran usando líneas discontinuas en la división descendente de la rama más a la derecha que tiene su origen en el nodo raíz del árbol de división 92.

En comparación con la estructura de CU en HEVC y en comparación con la estructura QTBT, la estructura de árboles de tipos múltiples ilustrada en las FIGS. 7A y 7B proporciona una mejor eficiencia de codificación, ya que las divisiones de bloques son más flexibles. Además, la introducción del árbol ternario del lado central proporciona una localización más flexible de las señales de vídeo. Para mantener la precisión con respecto a las partes divididas de PT de un bloque que se divide según la estructura de árboles de tipos múltiples, el codificador de vídeo 20 puede señalar, para cada nodo de división en las partes de PT, una indicación del tipo de división implementado con respecto a ese nodo de división en particular.

Como soporte a la precisión con respecto a las partes divididas de PT, el codificador de vídeo 20 puede señalar información que indica la estructura de árbol (árbol binario o ternario del lado central) así como la orientación de la división (horizontal o vertical). Por ejemplo, el codificador de vídeo 20 puede usar una asignación de palabras de código para indicar la estructura de árbol y la orientación de la misma para la división de cada nodo de división de las partes divididas de PT del bloque. La FIG. 7B incluye valores decimales de palabras de código que el codificador de vídeo 20 puede señalar para cada tipo de división que es posible dentro de la parte de PT de una estructura de división de árboles de tipos múltiples. Como se muestra en la FIG. 7B, el valor decimal de siete (7) indica división basada en árbol ternario del lado central vertical, el valor decimal de cinco (5) indica división basada en árbol ternario del lado central horizontal, el valor decimal de seis (6) indica división basada en árbol binario vertical y el valor decimal de cuatro (4) indica división basada en árbol binario horizontal. El valor decimal de cero (0) identifica un nodo hoja en la parte dividida de PT del bloque. Es decir, el codificador de vídeo 20 puede señalar la palabra de código que representa el valor decimal de cero (0) para indicar que el sub-bloque correspondiente no se divide más.

La FIG. 8 es un diagrama conceptual que ilustra ejemplos de palabras de código, expresadas en formato binario, que el codificador de vídeo 20 puede señalar para los diversos esquemas de división que son posibles en la parte de PT de divisiones de bloques divididos en árboles de tipos múltiples. La FIG. 8 también ilustra los esquemas de división correspondientes para las diversas palabras de código. Los valores binarios ilustrados en la FIG. 8 corresponden a los valores decimales ilustrados en la FIG. 7B. De este modo, la palabra de código binaria "111" indica división basada en árbol ternario del lado central vertical, la palabra de código binaria "101" indica división basada en árbol ternario del lado central horizontal, la palabra de código binaria "110" indica división basada en árbol binario vertical y la palabra de código binaria "100" indica división basada en árbol binario horizontal. La palabra de código binaria "000" identifica un nodo hoja en la parte dividida de PT del bloque. El primer contenedor (más a la izquierda) representa el valor de un indicador de división de PT (que indica si la CU se divide o no), el segundo contenedor representa la dirección de división de PT (p. ej., horizontal o vertical) y el tercer contenedor representa el modo de división de PT (p. ej., árbol binario o árbol ternario del lado central).

Las FIGS. 7B y 8 ilustran un ejemplo de asignación de palabras de código para la señalización de información de división de la parte de PT según la tecnología de división basada en árboles de tipos múltiples existente. Como se muestra, según la estructura de división de árboles de tipos múltiples, el codificador de vídeo 20 puede señalar tres bits o contenedores para cada nodo PT para indicar la información de división para el nodo PT correspondiente. A la inversa, el decodificador de vídeo 30 puede decodificar tres bits o contenedores para determinar la división de bloques en cada nodo PT. De nuevo, los diversos esquemas de división de bloques representados por las distintas palabras de código mostradas en la FIG. 8 son no dividido, árbol binario horizontal, árbol binario vertical, árbol ternario horizontal y árbol ternario vertical. En diversos escenarios de casos de uso, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden procesar un número importante de palabras de código con respecto a un único bloque, como en los casos en que la parte de PT de la estructura de árboles de tipos múltiples comienza pronto y/o persiste a través de la profundidad de división máxima. Dicho ejemplo se ilustra en las FIGS. 7A y 7B. La estructura de árboles de tipos múltiples tiene una naturaleza en dos niveles (RT y PT), y la profundidad de árbol para un bloque raíz indica un intervalo de tamaños de bloque para los nodos hoja.

Así, la estructura de división de árboles de tipos múltiples, a la vez que resulta beneficiosa desde los puntos de vista de la precisión de codificación y la flexibilidad de división, puede ser en muchos escenarios un esquema de codificación con muchos recursos y mucho ancho de banda. Las características de estructura de árboles de tipos múltiples expuestas anteriormente, junto con el añadido de divisiones ternarias por medio de la estructura de árbol ternario del

lado central, aumentan el número de bits/contenedores requeridos para señalar información de división de la parte de PT según la estructura de árboles de tipos múltiples. A modo de ejemplo, para contenido de vídeo regular codificado según la estructura de árboles de tipos múltiples, del nueve al catorce por ciento (9%-14%) de la cantidad total de bits del flujo de bits se consume para señalar palabras de código según la asignación de palabras de código ilustradas en la FIG. 8. Otra posible cuestión que presentan las tecnologías descritas anteriormente es que la modelización de contexto de la señalización de tipo árbol usa principalmente la profundidad relativa entre el bloque o bloques vecinos y el bloque actual para determinar la probabilidad de una división adicional para el bloque actual.

Esta descripción se dirige a técnicas que abordan (p. ej., mitigando y, en algunos casos, eliminando potencialmente) las cuestiones descritas anteriormente, así como otras cuestiones presentadas por las tecnologías y propuestas existentes de codificación de vídeo posteriores a HEVC. Las técnicas de esta descripción se dirigen generalmente a aliviar el consumo de recursos y ancho de banda de la estructura de división de árboles de tipos múltiples, a la vez que se continúan aprovechando los beneficios proporcionados por la estructura de división de árboles de tipos múltiples. Esta descripción proporciona diversas técnicas para mejorar la eficiencia de uso de la estructura de árboles de tipos múltiples, y las técnicas de esta descripción pueden implementarse individualmente o en diversas combinaciones y/o secuencias. De este modo, determinados aspectos de esta descripción se describen como dirigidos a una codificación más eficiente de los tipos de árbol.

Según algunos ejemplos de la presente descripción, en ciertas condiciones, el requisito de tres contenedores para señalar la información de división de PT puede exceptuarse en lo que se refiere a información que el codificador de vídeo 20 señala al decodificador de vídeo 30 para división de la parte de PT. En algunos ejemplos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden determinar información de división de PT para un bloque basándose en información disponible de bloque o bloques vecinos ya codificados, reduciendo así el número de palabras de código que se señalizan. En otros ejemplos, el codificador de vídeo 20 puede, según los aspectos de la presente descripción, señalar un esquema diferente de palabras de código a partir del esquema mostrado en la FIG. 8. A continuación se describen en mayor detalle varios ejemplos de las restricciones de codificación que pueden implementarse según las técnicas de esta descripción. Algunas técnicas de ejemplo de esta descripción se dirigen al uso de contextos más fiables para una modelización de contexto mejor (p. ej., potenciada).

Algunas técnicas de esta descripción se basan en un reconocimiento de que el tamaño de un bloque (un bloque "actual") con respecto al tamaño de bloque o bloques vecinos del bloque actual puede usarse para seleccionar los contextos para la codificación de tipo de árbol de PT del bloque actual. Por ejemplo, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden implementar una o más de las técnicas descritas en esta invención para determinar si un bloque actual debe o no dividirse adicionalmente, y puede basar la determinación en una relación entre el tamaño del bloque actual y uno o más de sus bloques vecinos. Usando la relación entre el tamaño del bloque actual y el o los tamaños del bloque o bloques vecinos para determinar si el bloque actual debe dividirse adicionalmente, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden reducir la carga adicional de señalización para codificación basada en árboles de tipos múltiples, ya que no es necesario señalar el indicador de división de PT (el contenedor más a la izquierda de las palabras de código ilustradas en la FIG. 8) para el bloque actual. En su lugar, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden realizar operaciones similares entre sí para determinar si se debe dividir o no el bloque actual basándose en el tamaño relativo en comparación con el bloque o bloques vecinos, eliminando así la necesidad de señalización explícita del indicador de división de PT.

En un ejemplo, basándose en una determinación de que la anchura del bloque actual es mayor que la anchura de un bloque vecino superior, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden determinar que el bloque actual probablemente debe dividirse adicionalmente. Es decir, si el bloque vecino superior tiene una anchura menor que el bloque actual, entonces el codificador de vídeo 20 y el decodificador de vídeo 30 pueden inferir que el bloque vecino superior es el resultado de una división, y que el bloque actual también debe dividirse. De este modo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden aprovechar la información de bloques vecinos superiores codificados previamente para reducir la carga adicional de señalización con respecto a la información de división para el bloque actual, mientras realizan también la determinación de si debe dividirse el bloque actual.

De forma similar, basándose en una determinación de que la altura del bloque actual es mayor que la altura de un bloque vecino izquierdo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden determinar que el bloque actual probablemente debe dividirse adicionalmente. Es decir, si el bloque vecino izquierdo tiene una altura menor que el bloque actual, entonces el codificador de vídeo 20 y el decodificador de vídeo 30 pueden inferir que el bloque vecino izquierdo es el resultado de una división, y que el bloque actual también debe dividirse. De este modo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden aprovechar la información de bloques vecinos izquierdos codificados previamente para reducir la carga adicional de señalización con respecto a la información de división para el bloque actual, mientras realizan también la determinación de si debe dividirse el bloque actual.

Además, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden usar el tamaño relativo de uno o más entre un bloque vecino superior izquierdo, superior derecho o inferior izquierdo en comparación con el tamaño del bloque actual para inferir si el bloque actual debe dividirse adicionalmente. Por ejemplo, si el codificador de vídeo 20 y el decodificador de vídeo 30 determinan que el área del bloque actual es mayor que el área de uno de entre el bloque

vecino superior izquierdo, superior derecho o inferior izquierdo, entonces el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden determinar que el bloque actual probablemente debe dividirse adicionalmente. En diversas partes de la presente descripción, el área del bloque puede referirse como "tamaño" del bloque y la información del área relativa puede referirse como "tamaño relativo" o "relación de tamaño".

Las determinaciones de anchura relativa, altura relativa y área relativa se refieren en esta invención como "eventos". En algunos ejemplos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden usar un número agregado de ocurrencias de uno o más de los eventos enumerados anteriormente para determinar el contexto para el indicador de división de PT, p. ej., una inferencia de si el bloque actual debe dividirse adicionalmente. En algunos ejemplos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden usar un evento individual para formar un conjunto de contextos para el indicador de división de PT.

En algunos ejemplos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden utilizar la anchura del bloque vecino superior y la altura del bloque vecino izquierdo para crear un contexto para una dirección de división de PT. Es decir, según estos ejemplos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden inferir el valor del contenedor medio de las palabras de código ilustradas en la FIG. 8, basándose en la anchura del bloque vecino superior y la altura del bloque vecino izquierdo. Si la anchura del bloque vecino superior es menor que la anchura del bloque actual y la altura del bloque vecino izquierdo es mayor o igual que la altura del bloque actual, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden determinar que el bloque actual probablemente debe dividirse en sentido vertical.

De forma similar, si la altura del bloque vecino izquierdo es menor que la altura del bloque actual y la anchura del bloque vecino superior es mayor que o igual que la anchura del bloque actual, entonces el codificador de vídeo 20 y el decodificador de vídeo 30 pueden determinar que el bloque actual probablemente debe dividirse en sentido horizontal. Según estos aspectos de la presente descripción, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementar las técnicas de esta descripción para inferir la dirección de división de PT basándose en información accesible desde bloques vecinos codificados previamente. De este modo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden reducir el número de instancias en que debe señalizarse la dirección de división de PT en la forma del esquema del contenedor medio de la palabra de código ilustrado en la FIG. 8.

Según algunos ejemplos de la presente descripción, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden utilizar la anchura del bloque vecino superior y la altura del bloque vecino izquierdo para inferir el contexto para el modo de división de PT. Es decir, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden utilizar la anchura del bloque vecino superior y la altura del bloque vecino izquierdo para seleccionar entre la estructura de árbol binario y la estructura de árbol ternario del lado central para la división del bloque actual, si el bloque actual debe ser dividido en la parte de PT. En estos ejemplos, si la anchura del bloque vecino superior es menor que la anchura del bloque actual y el bloque actual se divide verticalmente, entonces el codificador de vídeo 20 y el decodificador de vídeo 30 pueden determinar que el bloque actual debe dividirse según la estructura de división de árbol ternario del lado central.

En estos ejemplos, si la altura del bloque vecino izquierdo es menor que la altura del bloque actual y el bloque actual se divide horizontalmente, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden determinar que el bloque actual probablemente debe dividirse según la estructura de división de árbol ternario del lado central. De esta manera, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementar las técnicas de esta descripción para inferir el valor del contenedor más a la derecha de las palabras de código ilustradas en la FIG. 8, basándose en información de bloques vecinos codificados previamente.

Según algunos ejemplos de la presente descripción, si el codificador de vídeo 20 y el decodificador de vídeo 30 determinan que los bloques vecinos del bloque actual no están disponibles, entonces el codificador de vídeo 20 y el decodificador de vídeo 30 pueden usar un valor de contexto por omisión durante la derivación de un contexto actual. Por ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden aplicar el contexto por omisión con respecto a codificación CABAC o codificación de entropía de un elemento sintáctico que indica la información de división en relación con el bloque actual.

En algunos ejemplos, si el codificador de vídeo 20 y el decodificador de vídeo 30 determinan que están permitidas divisiones de RT o PT diferentes para distintos componentes (como componentes Y, Cb, Cr o de profundidad), entonces el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden aplicar todas las técnicas mencionadas anteriormente (p. ej., todas las técnicas descritas anteriormente con respecto al uso de la altura/anchura/área relativas), pero usando los bloques asociados en los otros componentes. Es decir, en lugar de usar información de los bloques vecinos expuesta anteriormente, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden usar la altura/anchura/área relativas con respecto a los componentes Y, Cb, Cr o de profundidad de un bloque. Como se usa en esta invención, 'Y' denota un componente de luminancia, Cb denota un componente de crominancia y Cr denota otro componente de crominancia.

Diversas técnicas de esta descripción se basan en un reconocimiento de que cuando se calcula el contexto de los tipos de árboles de PT, las posiciones de bloques vecinos pueden definirse de manera estática o dinámica para

adaptarse a las señales de vídeo de diversas características. Por ejemplo, estas técnicas permiten que el codificador de vídeo 20 y el decodificador de vídeo 30 seleccionen un bloque vecino en particular, con el fin de realizar una o más de las técnicas basadas en altura/anchura/área relativas descritas anteriormente. A modo de ejemplo, si el bloque actual tiene múltiples bloques vecinos superiores, entonces el codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementar las técnicas descritas en esta invención para seleccionar uno de los bloques vecinos superiores con el que realizar determinaciones basadas en la anchura relativa. De forma similar, si el bloque actual tiene múltiples bloques vecinos izquierdos, entonces el codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementar las técnicas descritas en esta invención para seleccionar uno de los bloques vecinos izquierdos con el que realizar determinaciones basadas en la altura relativa.

La FIG. 9 es un diagrama conceptual que ilustra posiciones candidatas de varias ubicaciones vecinas superiores y ubicaciones vecinas izquierdas para modelización de contexto de un indicador de división de PT, según los aspectos de la presente descripción. Como el rango de la relación de aspecto de las CU se vuelve más alto en comparación con las relaciones de aspecto proporcionadas por las normas de codificación anteriores, la posición del bloque vecino superior puede elegirse entre posiciones como la inmediatamente encima de la esquina superior izquierda del bloque actual, inmediatamente encima del bloque 4x4 central a lo largo del límite superior del bloque actual o inmediatamente encima del bloque 4x4 más a la derecha en la esquina superior derecha del bloque actual.

Las posiciones T0-T3 en la FIG. 9 ilustran ejemplos de posiciones candidatas para el bloque vecino superior que se usarán para determinaciones basadas en la anchura relativa. De forma similar, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementar las técnicas de esta descripción para seleccionar el bloque vecino izquierdo entre posiciones como la inmediatamente a la izquierda de la esquina superior izquierda del bloque actual, la inmediatamente a la izquierda del bloque 4x4 central a lo largo del límite izquierdo del bloque actual o la inmediatamente a la izquierda del bloque 4x4 más abajo en la esquina inferior izquierda del bloque actual. Las posiciones L0-L3 en la FIG. 9 ilustran ejemplos de posiciones candidatas para el bloque vecino izquierdo que se usarán para determinaciones basadas en la altura relativa. La selección de los bloques vecinos superior e izquierdo según las técnicas descritas en esta invención puede producir una mejora en la exactitud cuando aumenta la relación de aspecto de los tamaños de bloque.

En un ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden elegir un bloque colocado en una ubicación entre T0 y T3 como el bloque vecino superior, y pueden elegir un bloque colocado en una ubicación entre L0 y L3 como el bloque vecino izquierdo cuando realizan modelización de contexto para el bloque actual. En otro ejemplo, el codificador de vídeo 20 puede especificar la posición del bloque vecino superior y el bloque vecino izquierdo señalizando la información en el conjunto de parámetros de secuencia (Sequence Parameter Set, SPS), el conjunto de parámetros de imagen (Picture Parameter Set, PPS) o el encabezamiento de fragmento (Slice Header, SH). En estos ejemplos, el decodificador de vídeo 30 puede seleccionar la posición del bloque vecino superior y el bloque vecino izquierdo usando información señalizada en el SPS, el PPS o el SH. Usando la información de posición del bloque vecino señalizada en el SPS/PPH/SH, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden usar la misma información para selección del bloque vecino en todos los bloques de división de PT de una secuencia, una imagen o un fragmento. De este modo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden reducir la carga adicional de señalización reduciendo el número de palabras de código individuales que deben ser señalizadas para la división basada en PT.

En un ejemplo, el codificador de vídeo 20 puede señalar un par o tupla 2 seleccionado entre el conjunto {(TL, TR), (TL, BL), (TR, BL)} del SPS, el PPS o el SH para modelización de contexto. En un ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden elegir estáticamente un par o tupla 2 seleccionado entre el conjunto {(TL, TR), (TL, BL), (TR, BL)}. En la notación usada anteriormente, 'TL' representa un bloque vecino superior izquierdo, 'TR' representa un bloque vecino superior derecho, 'BL' representa un bloque vecino inferior izquierdo y 'BR' representa un bloque vecino inferior derecho.

En algunos ejemplos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden elegir la posición del bloque vecino superior y el bloque vecino izquierdo basándose en la forma del bloque actual. Por ejemplo, si la anchura del bloque actual es mayor que la altura, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden elegir T1 o T2 ilustrados en la FIG. 9 como el bloque vecino superior que se usará en determinaciones basadas en la anchura relativa de la presente descripción. En caso contrario (p. ej., si la anchura del bloque actual es menor o igual que la altura), el codificador de vídeo 20 y el decodificador de vídeo 30 pueden elegir T0 ilustrado en la FIG. 9 como el bloque vecino superior que se usará en determinaciones basadas en la anchura relativa de la presente descripción. De forma similar, si la altura del bloque actual es mayor que la anchura del bloque actual, entonces el codificador de vídeo 20 y el decodificador de vídeo 30 pueden elegir L1 o L2 ilustrados en la FIG. 9 como el bloque vecino izquierdo para determinaciones basadas en la altura relativa de la presente descripción. En caso contrario, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden elegir L0.

Según algunos ejemplos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden modelizar un contexto usando una combinación de información de bloques vecinos en todas las posiciones mostradas en la FIG. 9, si las posiciones son disjuntas. Por ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden comparar la

anchura, altura o tamaño (p. ej., área) del bloque actual con la métrica correspondiente para todos los bloques vecinos circundantes ilustrados en la FIG. 9, al objeto de predecir el valor de indicador de la división de PT, la dirección de división de PT o el modo de división de PT. Usando una combinación de la información de múltiples bloques vecinos de este modo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden predecir con más precisión el indicador de división de PT, la dirección de división de PT y el modo de división de PT, debido a la mayor cantidad de datos heurísticos considerados.

En otro ejemplo, el codificador de vídeo 20 puede aplicar una restricción según la cual no se usa señalización adicional. Por ejemplo, si el bloque vecino izquierdo y el bloque vecino superior están los dos divididos verticalmente según la estructura de árbol binario, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden inferir que el bloque actual debe dividirse verticalmente según la estructura de árbol binario. Otras de estas restricciones pueden aplicarse según los aspectos de la presente descripción, y se observará que las restricciones de esta descripción no se limitan al ejemplo de la restricción de división binaria vertical expuesta anteriormente.

Según algunos aspectos de la presente descripción, el codificador de vídeo 20 puede aplicar restricciones que pueden ser apropiadas caso por caso (p. ej., descritas también como restricciones 'propias') en la señalización de tipo árbol. En algunos casos, el codificador de vídeo 20 puede omitir los contenedores usados para señalar la dirección de división de PT o el modo de división de PT, dada una profundidad de RT y PT admisible. Con respecto a la exposición de estas técnicas la anchura mínima de bloque y la altura mínima de bloque se denotan por mW y mH, respectivamente.

En un ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden determinar que, si la anchura del bloque actual es igual a mW, entonces solo puede producirse división de PT horizontal en relación con el bloque actual. En este caso, el codificador de vídeo 20 no puede señalar la dirección de división de PT, ya que la dirección de división de PT es inferida por el decodificador de vídeo 30 en relación con el bloque actual. De forma similar, en un ejemplo, si la altura del bloque actual es igual a mH, entonces el codificador de vídeo 20 no puede señalar la dirección de división de PT, basándose en el decodificador de vídeo 30 que infiere que la dirección de división de PT debe ser horizontal.

En un ejemplo, si la anchura del bloque actual es igual a $(2 \times mW)$ y el bloque actual se divide verticalmente, entonces el codificador de vídeo 20 no puede señalar el modo de división de PT, ya que en este caso solo se permite división de árbol binario vertical. De forma similar, si la altura del bloque actual es igual a $(2 \times mH)$ y el bloque actual se divide horizontalmente, entonces el codificador de vídeo 20 no puede señalar el modo de división de PT, ya que en este caso solo se permite división de árbol binario horizontal.

En algunos ejemplos, el codificador de vídeo 20 puede señalar una indicación 'deshabilitado' con respecto al árbol binario o al árbol ternario o a ambos en el flujo de bits. Por ejemplo, el codificador de vídeo 20 puede señalar la indicación 'deshabilitado' en una o más estructuras sintácticas, como un SPS y/o un PPS y/o un encabezamiento de fragmento. Si un tipo de árbol se señala como deshabilitado en un cierto nivel, entonces el codificador de vídeo 20 puede omitir la señalización del modo de división de PT dentro de ese nivel. Por ejemplo, si el codificador de vídeo 20 señala un estado deshabilitado en el PPS (es decir, en el nivel de imagen), entonces el codificador de vídeo puede omitir la señalización del modo de división de PT para toda la imagen.

En otro ejemplo más, si están permitidas divisiones de RT o PT diferentes para distintos componentes (como componentes Y, Cb, Cr o de profundidad) del bloque actual, el codificador de vídeo 20 puede aplicar la restricción de división de RT o PT para el componente actual según la división de RT y/o PT de los bloques asociados en los otros componentes. Por ejemplo, cuando se codifica un bloque en componentes de Cb o Cr, si el bloque de componente Y se divide verticalmente, entonces el codificador de vídeo 20 no puede señalar el modo de división horizontal para el componente actual.

Según algunos aspectos de la presente descripción, el codificador de vídeo 20 puede seleccionar una palabra de código de una tabla de búsqueda de codificación de longitud variable (Variable Length Coding, VLC), en lugar de señalar las palabras de código ilustradas en la FIG. 8. Las palabras de código de la tabla de búsqueda VLC pueden proporcionar una flexibilidad alternativa para algunas señales de vídeo donde la probabilidad conjunta de cada tipo de división sigue una distribución sesgada o altamente sesgada. El procedimiento de binarización en sí puede encapsular la información de codificación de los bloques vecinos, para producir la distribución de probabilidad sesgada. En algunos ejemplos de la presente descripción, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden usar diferentes procedimientos de binarización para la división del bloque actual, según diferentes divisiones de bloques vecinos.

En un ejemplo, el codificador de vídeo 20 puede usar códigos de longitud fija (Fixed Length, FL) de dos contenedores para señalar eventos específicos de división. Como se expuso anteriormente, son posibles cuatro tipos diferentes de división de PT dentro de la estructura de árboles de tipos múltiples. De nuevo, los cuatro tipos de división de PT son árbol binario horizontal, árbol binario vertical, árbol ternario horizontal del lado central y árbol ternario vertical del lado central. En varios ejemplos de los códigos FL de dos contenedores de la presente descripción, la definición de los contenedores primero y segundo pueden ser intercambiables. Es decir, el primer contenedor puede representar la

dirección de división y el segundo contenedor puede representar el tipo de árbol, o a la inversa.

Alternativamente, el codificador de vídeo 20 puede usar códigos unarios truncados (Truncated Unary, TU) para señalar las divisiones. La Tabla 1 mostrada a continuación recoge un ejemplo de un esquema de asignación de palabras de código que el codificador de vídeo 20 puede usar, según los aspectos de la presente descripción.

Tabla 1. Ejemplo de asignación de palabras de código para cada tipo de división en PT

División	Palabra de código
Árbol binario horizontal	10
Árbol binario vertical	0
Árbol ternario horizontal del lado central	110
Árbol ternario vertical del lado central	111

Se observará que la asignación de palabras de código para cada tipo de división puede basarse en la probabilidad de cada división, y la Tabla 1 anterior simplemente sirve como un ejemplo no limitativo. El codificador de vídeo 20 puede asignar la palabra de código más corta al escenario que se encuentra con mayor frecuencia. En el ejemplo de la Tabla 1, la división de árbol binario vertical puede representar el escenario que se encuentra con mayor frecuencia en la parte dividida de PT, y de este modo, el codificador de vídeo 20 puede asignar la palabra de código '0' para indicar una división de árbol binario vertical. En muchos escenarios de casos de uso, más del 80% de los bloques con división de PT en un flujo se dividen de la misma forma. De este modo, el codificador de vídeo 20 puede reducir los requisitos de tasa de bits para división basada en árboles de tipos múltiples, asignando palabras de código más cortas a las indicaciones señalizadas con mayor frecuencia.

En un ejemplo, si el bloque vecino izquierdo y el bloque vecino superior están los dos divididos verticalmente usando un árbol binario, es altamente posible o probable que el codificador de vídeo 20 y el decodificador de vídeo 30 puedan determinar que el bloque actual sea dividido también verticalmente. En este caso, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden usar una palabra de código más corta para división binaria vertical. En otras palabras, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden usar diferentes procedimientos de binarización de divisiones, dependiendo de las divisiones de bloques vecinos, según este ejemplo. De este modo, el codificador de vídeo 20 puede reducir los requisitos de tasa de bits para división basada en árboles de tipos múltiples, asignando palabras de código más cortas a la indicación que será señalizada más probablemente para el bloque actual.

En un ejemplo, el codificador de vídeo 20 puede señalar la selección de palabra de código en diversas estructuras sintácticas del flujo de bits, como un SPS, un PPS o un SH. En otro ejemplo, cuando están permitidas divisiones de RT o PT diferentes para distintos componentes (como componentes Y, Cb, Cr o de profundidad), la selección de la tabla de búsqueda VLC para el componente actual puede aplicarse según la división de RT y/o PT de los bloques asociados en los otros componentes. Por ejemplo, cuando se codifica un bloque en componentes Cb o Cr y el bloque asociado en componente Y se divide verticalmente, el codificador de vídeo 20 puede señalar una palabra de código acortada para la división basada en árbol binario vertical. En otras palabras, según este ejemplo, el codificador de vídeo 20 puede usar diferentes procedimientos de binarización de divisiones, dependiendo del bloque asociado divisiones en el otro componente.

Según determinadas técnicas de la presente descripción, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementar una manera predefinida de división para todas las CTU que se extienden en un límite de imagen. Como la estructura de división de CTU de imágenes cruzadas está predefinida en estos ejemplos, el codificador de vídeo 20 no puede señalar la información de tipo de árbol hasta que la división produzca como resultado que todas las muestras de cada CU estén situadas dentro de la imagen actual.

En un ejemplo, la CTU puede dividirse recursivamente por división de RT sin que el codificador de vídeo 20 señale ningún indicador de división de CU, hasta que la división de RT recursiva alcance una fase en que todos los nodos representan una CU resultante, cuya totalidad está situada dentro de la imagen actual. En otros ejemplos, la CTU puede dividirse recursivamente usando un tipo seleccionado de división de PT sin que el codificador de vídeo 20 señale ningunos de los indicadores de división de CU, hasta que la división recursiva alcance una fase en que todos los nodos representan una CU resultante cuya totalidad está situada dentro de la imagen actual. En un ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden dividir la CTU recursivamente buscando la representación más favorable usando la señalización regular de tipo árbol, que incluye tanto RT como PT. Por ejemplo, para una CTU para la cual una mitad derecha se sitúa fuera de la imagen, una división de RT de un nivel forma una división válida. Además, ninguna división en RT con una división de PT de un nivel forma otra división válida.

Una división válida se define como la división donde la totalidad de cada CU respectiva está situada dentro de la imagen actual. En otro ejemplo, el codificador de vídeo 20 puede dividir la CTU adaptativamente, y el codificador de vídeo 20 puede señalar un indicador de un bit, o un índice, o múltiples indicadores para permitir que el decodificador de vídeo 30 decida o determine cuál de los esquemas de división mencionados anteriormente debería implementarse. En este ejemplo, el decodificador de vídeo 30 puede usar la información recibida para seleccionar un esquema de división que corresponda a la división adaptativa implementada por el codificador de vídeo 20. De este modo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden reducir los requisitos de tasa de bits para señalización de palabras de código en casos de una CTU que abarca un límite de imagen. Como el mecanismo de división recursiva está predefinido para dividir dicha CTU en CU cada una de las cuales se sitúa totalmente dentro de la imagen actual, el codificador de vídeo 20 no necesita señalar repetidamente las palabras de código para indicar la información de división en cada nodo, al menos hasta que cada CU esté contenida dentro de una sola imagen.

De nuevo, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden implementar las técnicas descritas anteriormente de forma individual, o en cualquier combinación en que dos o más de las técnicas puedan combinarse lógicamente. Como se describió anteriormente, las técnicas de esta descripción son aplicables a escenarios en los que se usan varios tipos de árboles para división de bloques. Por ejemplo, la división de árboles de región (RT) puede incluir pero no se limita a una división de árbol cuaternario, y la división de árboles de predicción (PT) puede incluir pero no se limita a una división de árbol binario y/o una división de árbol ternario simétrico del lado central.

A continuación se describen varias implementaciones de las técnicas descritas anteriormente. Por ejemplo, con respecto a las técnicas basadas en la selección del bloque vecino descritas anteriormente con respecto a la FIG. 9, para cada nodo en la parte del árbol de predicción (PT), el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden realizar la siguiente codificación:

1. Para el indicador de división de PT, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden usar los tamaños de píxel de la anchura, la altura y el área para deducir la información de contexto.

a) En una implementación de ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden seleccionar la anchura del bloque a partir del bloque vecino superior situado en la posición T2 (ilustrada en la FIG. 9). En esta implementación, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden seleccionar la altura de bloque a partir del bloque vecino izquierdo situado en la posición L2, y pueden seleccionar los tamaños de bloque (p. ej., áreas) a partir del par de (TL, TR). Sean la anchura y la altura del bloque W y H, respectivamente. El valor de índice de contexto puede calcularse como:

$$CTX = (W > W_{T2}) + (H > H_{L2}) + (W * H > STL) + (W * H > STR) \quad (1)$$

donde STL y STR representan los tamaños de bloques que ocupan las ubicaciones de TL y TR en la FIG. 9. Según la Ecuación (1), puede seleccionarse un contexto adicional si se cumple la condición siguiente:

$$CTX = ((W < W_{T2}) \& \& (H < H_{L2}) \& \& (W * H < STL) \& \& (W * H < STR)) ? 5 : CTX \quad (2)$$

En esta implementación, las posiciones de T0, T1, T2 y T3 se calculan como:

$$T0 = (X0, Y0 - 1) \quad (3)$$

$$T1 = \left(X0 + \frac{W}{2} - 1, Y0 - 1 \right)$$

$$T2 = \left(X0 + \frac{W}{2}, Y0 - 1 \right)$$

$$T3 = (X0 + W - 1, Y0 - 1)$$

De forma similar, las posiciones de L0, L1, L2 y L3 se calculan como:

$$\begin{aligned}
 L0 &= (X0 - 1, Y0) \\
 L1 &= \left(X0 - 1, Y0 + \frac{H}{2} - 1 \right) \\
 L2 &= \left(X0 - 1, Y0 + \frac{H}{2} \right) \\
 L3 &= (X0 - 1, Y0 + H - 1)
 \end{aligned}
 \tag{4}$$

donde (X0, Y0) es la coordenada de píxeles de la esquina superior izquierda del bloque actual. Los tamaños de unidad en las Ecuaciones (3) y (4) son iguales a la anchura mínima de bloque y a la altura mínima de bloque para las direcciones X e Y, respectivamente.

b) En una implementación de ejemplo, puede definirse un conjunto de contextos más pequeño del modo siguiente:

$$\begin{aligned}
 CTX &= (W > W_{T2}) + (H > H_{L2}) + (W * H > STL) + (W * H > STR) \\
 CTX &= ((W < W_{T2}) \& \& (H < H_{L2}) \& \& (W * H < STL) \& \& (W * H < STR)) ? 0 : CTX \\
 CTX &= (CTX \geq 3) ? 3 : CTX
 \end{aligned}
 \tag{5}$$

c) En una implementación de ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden seleccionar la anchura del bloque a partir de una cualquiera de las posiciones superiores entre T0 y T3, y pueden seleccionar la altura de bloque a partir de una cualquiera de las posiciones entre L0 y L3. En esta implementación, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden seleccionar un par de tamaños de bloques vecinos a partir de los siguientes conjuntos:

$$\{(TL, TR), (TR, BL), (TL, BL)\}$$

El valor de índice de contexto puede calcularse usando las Ecuaciones (1) y (2) anteriores, pero con las posiciones de T2, L2, STL y STR sustituidas por las seleccionadas.

Alternativamente, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden hacer la selección basándose en el conjunto de posiciones que minimiza la entropía condicional, y el índice del conjunto de posiciones puede señalizarse en el flujo de bits, por ejemplo a través de SPS, PPS o encabezamientos de fragmento.

2. Para una dirección de división de PT, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementar las técnicas de esta descripción para utilizar la anchura del bloque, la altura y sus valores relativos a los vecinos superior e izquierdo para crear el conjunto de contextos.

a) En una implementación de ejemplo, un conjunto de modelos de contexto puede definirse del modo siguiente (según la Tabla 2 mostrada a continuación):

Tabla 2

Condiciones	CTX
blkSizeX == blkSizeY	0
(blkSizeX > blkSizeY) && (blkSizeX <= 2*blkSizeY)	1
(blkSizeY > blkSizeX) && (blkSizeY <= 2*blkSizeX)	2
blkSizeX > 2*blkSizeY	3
blkSizeY > 2*blkSizeX	4

b) En una implementación de ejemplo, un conjunto de modelos de contexto puede definirse del modo siguiente (véase también la Tabla 3 mostrada a continuación). Sean **condL** y **condT** dos condiciones para los vecinos izquierdo y superior:

5 **condL** = (blkSizeX == blkSizeY) && (leftAvailable && blkSizeY > leftBlkSizeY)
 condT = (blkSizeX == blkSizeY) && (topAvailable && blkSizeX > topBlkSizeX)

Tabla 3

Condiciones	CTX
(blkSizeX == blkSizeY) && ! condT && ! condL	0
((blkSizeX > blkSizeY) && (blkSizeX <= 2*blkSizeY)) condT	1
((blkSizeY > blkSizeX) && (blkSizeY <= 2*blkSizeX)) condL	2
blkSizeX > 2*blkSizeY	3
blkSizeY > 2*blkSizeX	4

c) En una implementación de ejemplo, un conjunto de modelos de contexto con menor complejidad puede definirse del modo siguiente (véase también la Tabla 4 mostrada a continuación):

15 Sean **condL** y **condT** dos condiciones del modo siguiente:
 condL = (blkSizeX == blkSizeY) && (leftAvailable && blkSizeY > leftBlkSizeY)
 condT = (blkSizeX == blkSizeY) && (topAvailable && blkSizeX > topBlkSizeX)

Tabla 4

Condiciones	CTX
(blkSizeX == blkSizeY) && ! condT && ! condL	0
(blkSizeX > blkSizeY) condT	1
(blkSizeY > blkSizeX) condL	2

3. Para el modo de división de PT, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementar determinadas técnicas de esta descripción para utilizar la anchura del bloque, la altura y la relación relativa entre los bloques superior e izquierdo para crear el conjunto de contextos.

a) En una implementación de ejemplo, un conjunto de contextos puede definirse del modo siguiente (véase también la Tabla 5 mostrada a continuación). Sean **condL** y **condT** dos condiciones para el bloque vecino izquierdo y el bloque vecino superior:

30 **condL** = (leftAvailable && blkSizeY > leftBlkSizeY) && (horizontal_split)
 condT = (topAvailable && blkSizeX > topBlkSizeX) && (vertical_split)

condT = (topAvailable && blkSizeX > topBlkSizeX) && (vertical_split)

Tabla 5

Condiciones	CTX
! condL && ! condT	0
condL condT	1

b) En una implementación de ejemplo, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden definir un contexto como la probabilidad del indicador en sí. Es decir, el valor de índice de contexto se establece en 0.

c) En una realización, el codificador de vídeo 20 puede establecer la codificación de este contenedor como libre de contexto, y la codificación de omisión se realiza para el indicador en el motor CABAC.

4. Puede deducirse un sistema equivalente contando individualmente el número de niveles de árboles de un cuarto de tamaño (que tienen lugar en el RT, y las divisiones primera y tercera en los árboles ternarios) y el número de niveles de otros tipos de árboles de PT.

a) En una implementación de ejemplo, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden definir un sistema de profundidad bidireccional del modo siguiente. Sea la profundidad del árbol de cuarto de tamaño y la profundidad del árbol de mitad de tamaño D_Q y D_H , respectivamente. El codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden calcular una profundidad D equivalente como:

$$D = 2 * D_Q + D_H \quad (6)$$

b) El codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden aplicar una definición similar a la anchura y la altura tomando los valores logarítmicos de la anchura y la altura. Con el sistema equivalente, por ejemplo, la Ecuación (1) anterior puede reescribirse como:

$$CTX = (\log(W) > \log(W_{T2})) + (\log(H) > \log(H_{L2})) + (D < D_{TL}) + (D < D_{TR}) \quad (7)$$

donde D_{TL} y D_{TR} son las profundidades equivalentes de los bloques que ocupan las posiciones superior izquierda y superior derecha, respectivamente.

La FIG. 10 es un organigrama que ilustra un procedimiento 100 de ejemplo que puede realizar el codificador de vídeo 20 (o los circuitos de procesamiento del mismo) según distintos aspectos de la presente descripción. El procedimiento 100 puede comenzar cuando los circuitos de procesamiento del codificador de vídeo 20 comparan un valor de una dimensión del bloque actual con un valor de una dimensión correspondiente de un bloque vecino para obtener información del valor de dimensión relativa (102). En un ejemplo, los circuitos de procesamiento del codificador de vídeo 20 pueden comparar la anchura del bloque actual con una anchura de un bloque vecino superior que está situado encima del bloque actual. En otro ejemplo, los circuitos de procesamiento del codificador de vídeo 20 pueden comparar la altura del bloque actual con una altura de un bloque vecino izquierdo que está situado a la izquierda del bloque actual. De este modo, la dimensión correspondiente del bloque vecino representa una medición o métrica que indica información del factor de forma del bloque vecino en una forma similar a la dimensión del bloque actual.

Basándose en el valor de dimensión relativa, los circuitos de procesamiento del codificador de vídeo 20 pueden determinar que el bloque actual debe ser dividido según la parte de PT de la estructura de división de árboles de tipos múltiples (104). En un ejemplo, los circuitos de procesamiento del codificador de vídeo 20 pueden determinar que el bloque actual debe ser dividido según la parte de PT de la estructura de árboles de tipos múltiples, basándose en una determinación de que la anchura del bloque actual es menor que la anchura del bloque vecino superior. En otro ejemplo, los circuitos de procesamiento del codificador de vídeo 20 pueden determinar que el bloque actual debe ser dividido según la parte de PT de la estructura de árboles de tipos múltiples, basándose en una determinación de que la altura del bloque actual es menor que la altura del bloque vecino izquierdo.

A su vez, los circuitos de procesamiento del codificador de vídeo 20 pueden dividir el bloque actual según la estructura de árbol ternario del lado central o la estructura de árbol binario (106). Como se expuso anteriormente, la división de la parte de PT sigue la estructura de árbol binario o la estructura de árbol ternario del lado central. En algunos ejemplos, los circuitos de procesamiento del codificador de vídeo 20 pueden heredar el tipo de árbol (binario o ternario del lado central) del bloque vecino que se usó para determinar la dimensión relativa. A su vez, los circuitos de procesamiento del codificador de vídeo 20 pueden señalar un flujo de bits de vídeo codificado al decodificador de vídeo 30 (108). Por ejemplo, los circuitos de procesamiento del codificador de vídeo 20 pueden usar una interfaz de red u otro hardware de comunicaciones del codificador de vídeo 20 para señalar el flujo de bits.

En los ejemplos en que el bloque vecino es un bloque vecino superior y el valor de dimensión relativa se basa en una comparación de anchura, para dividir el bloque actual, los circuitos de procesamiento del codificador de vídeo 20 pueden dividir verticalmente el bloque actual. En los ejemplos en que el bloque vecino es un bloque vecino izquierdo y el valor de dimensión relativa se basa en una comparación de altura, para dividir el bloque actual, los circuitos de procesamiento del codificador de vídeo 20 pueden dividir horizontalmente el bloque actual.

En algunos ejemplos, el bloque vecino es un bloque vecino en diagonal del bloque actual. El bloque vecino en diagonal puede incluir cualquiera entre un bloque vecino superior izquierdo que está situado encima y a la izquierda del bloque actual, un bloque vecino superior derecho que está situado encima y a la derecha del bloque actual o un bloque vecino inferior izquierdo que está situado debajo y a la derecha del bloque actual. En estos ejemplos, para comparar el valor

de la dimensión del bloque actual con el valor de la dimensión correspondiente del bloque vecino en diagonal, los circuitos de procesamiento del codificador de vídeo 20 pueden comparar, basándose en el bloque vecino que es el bloque vecino en diagonal, un área del bloque actual con un área del bloque vecino superior.

En algunos ejemplos, los circuitos de procesamiento del decodificador de vídeo 30 pueden comparar una anchura del bloque actual con una altura del bloque actual para determinar una forma del bloque actual. En estos ejemplos, para seleccionar el bloque vecino, los circuitos de procesamiento del decodificador de vídeo 30 pueden seleccionar el bloque vecino entre uno de la pluralidad de bloques vecinos izquierdos o la pluralidad de bloques vecinos superiores basándose en la forma determinada del bloque actual.

En algunos ejemplos, los circuitos de procesamiento del codificador de vídeo 20 pueden seleccionar el bloque vecino entre uno de (i) una pluralidad de bloques vecinos superiores que incluye un primer bloque vecino superior que está situado encima de un sub-bloque 4x4 situado más a la izquierda a lo largo de un límite superior del bloque actual, un segundo bloque vecino superior que está situado encima de un sub-bloque 4x4 medio a lo largo del límite superior del bloque actual y un tercer bloque vecino superior que está situado encima de un sub-bloque 4x4 situado más a la derecha a lo largo del límite superior del bloque actual, o (ii) una pluralidad de bloques vecinos izquierdos que incluye un primer bloque vecino izquierdo que está situado a la izquierda de un sub-bloque 4x4 superior a lo largo de un límite izquierdo del bloque actual, un segundo bloque vecino izquierdo que está situado a la izquierda de un sub-bloque 4x4 medio a lo largo del límite izquierdo del bloque actual y un tercer bloque vecino izquierdo que está situado a la izquierda de un sub-bloque 4x4 inferior a lo largo del límite izquierdo del bloque actual. En algunos ejemplos, los circuitos de procesamiento del codificador de vídeo 20 pueden usar hardware de comunicación del codificador de vídeo 20 para señalar, en uno entre un conjunto de parámetros de secuencia (SPS), un conjunto de parámetros de imagen (PPS) o un encabezamiento de fragmento (SH), una indicación del bloque vecino seleccionado para uno o más bloques en una unidad de datos de vídeo que corresponde al SPS, el PPS o el SH respectivo.

En algunos ejemplos, los circuitos de procesamiento del codificador de vídeo 20 pueden asignar una palabra de código respectiva de una pluralidad de palabras de código a una combinación de tipo de árbol de dirección respectiva de una pluralidad de combinaciones de tipos de árboles de dirección para la división del bloque actual. En estos ejemplos, la pluralidad de palabras de código incluye una palabra de código 0, una palabra de código 10, una palabra de código 110 y una palabra de código 111, y la pluralidad de combinaciones de tipos de árboles de dirección incluye una combinación de árbol binario horizontal, una combinación de árbol binario vertical, una combinación de árbol ternario del lado central horizontal y una combinación de árbol ternario del lado central vertical. En estos ejemplos, para dividir el bloque actual según la parte de PT del esquema de división basada en árboles de tipos múltiples, los circuitos de procesamiento del codificador de vídeo 20 pueden dividir el bloque actual según una combinación de tipo de árbol de dirección particular incluida en la pluralidad de combinaciones de tipos de árboles de dirección. Los circuitos de procesamiento del codificador de vídeo 20 pueden usar hardware de comunicación del codificador de vídeo 20 para señalar, en un flujo de bits de vídeo codificado, la palabra de código respectiva asignada a la combinación de tipo de árbol de dirección particular según la cual se divide el bloque actual.

La FIG. 11 es un organigrama que ilustra un procedimiento 120 de ejemplo que pueden realizar los circuitos de procesamiento del decodificador de vídeo 30 según distintos aspectos de la presente descripción. El procedimiento 120 puede comenzar cuando los circuitos de procesamiento del decodificador de vídeo 30 comparan un valor de una dimensión del bloque actual con un valor de una dimensión correspondiente de un bloque vecino para obtener información del valor de dimensión relativa (122). En un ejemplo, los circuitos de procesamiento del decodificador de vídeo 30 pueden comparar la anchura del bloque actual con una anchura de un bloque vecino superior que está situado encima del bloque actual, basándose en el bloque vecino que es un bloque vecino superior. En otro ejemplo, los circuitos de procesamiento del decodificador de vídeo 30 pueden comparar la altura del bloque actual con una altura de un bloque vecino izquierdo que está situado a la izquierda del bloque actual, basándose en el bloque vecino que es un bloque vecino izquierdo. De este modo, la dimensión correspondiente del bloque vecino representa una medición o métrica que indica información del factor de forma del bloque vecino en una forma similar a la dimensión del bloque actual.

Basándose en el valor de dimensión relativa, los circuitos de procesamiento del decodificador de vídeo 30 pueden determinar que el bloque actual debe ser dividido según la parte de PT de la estructura de división de árboles de tipos múltiples (124). En un ejemplo, los circuitos de procesamiento del decodificador de vídeo 30 pueden determinar que el bloque actual debe ser dividido según la parte de PT de la estructura de árboles de tipos múltiples, basándose en una determinación de que la anchura del bloque actual es menor que la anchura del bloque vecino superior. En otro ejemplo, los circuitos de procesamiento del decodificador de vídeo 30 pueden determinar que el bloque actual debe ser dividido según la parte de PT de la estructura de árboles de tipos múltiples, basándose en una determinación de que la altura del bloque actual es menor que la altura del bloque vecino izquierdo.

A su vez, los circuitos de procesamiento del decodificador de vídeo 30 pueden dividir el bloque actual según la estructura de árbol ternario del lado central o la estructura de árbol binario (126). Como se expuso anteriormente, la división de la parte de PT sigue la estructura de árbol binario o la estructura de árbol ternario del lado central. En algunos ejemplos, los circuitos de procesamiento del decodificador de vídeo 30 pueden heredar el tipo de árbol (binario

o ternario del lado central) del bloque vecino que se usó para determinar la dimensión relativa. En los ejemplos en que el bloque vecino es un bloque vecino superior y el valor de dimensión relativa se basa en una comparación de anchura, para dividir el bloque actual, los circuitos de procesamiento del decodificador de vídeo 30 pueden dividir verticalmente el bloque actual. En los ejemplos en que el bloque vecino es un bloque vecino izquierdo y el valor de dimensión relativa se basa en una comparación de altura, para dividir el bloque actual, los circuitos de procesamiento del decodificador de vídeo 30 pueden dividir horizontalmente el bloque actual.

En algunos ejemplos, el bloque vecino es un bloque vecino en diagonal del bloque actual. El bloque vecino en diagonal puede incluir cualquiera entre un bloque vecino superior izquierdo que está situado encima y a la izquierda del bloque actual, un bloque vecino superior derecho que está situado encima y a la derecha del bloque actual o un bloque vecino inferior izquierdo que está situado debajo y a la derecha del bloque actual. En estos ejemplos, para comparar el valor de la dimensión del bloque actual con el valor de la dimensión correspondiente del bloque vecino en diagonal, los circuitos de procesamiento del decodificador de vídeo 30 pueden comparar, basándose en el bloque vecino que es el bloque vecino en diagonal, un área del bloque actual con un área del bloque vecino superior.

En algunos ejemplos, los circuitos de procesamiento del decodificador de vídeo 30 pueden seleccionar el bloque vecino entre uno de (i) una pluralidad de bloques vecinos superiores que incluye un primer bloque vecino superior que está situado encima de un sub-bloque 4x4 situado más a la izquierda a lo largo de un límite superior del bloque actual, un segundo bloque vecino superior que está situado encima de un sub-bloque 4x4 medio a lo largo del límite superior del bloque actual y un tercer bloque vecino superior que está situado encima de un sub-bloque 4x4 situado más a la derecha a lo largo del límite superior del bloque actual, o (ii) una pluralidad de bloques vecinos izquierdos que incluye un primer bloque vecino izquierdo que está situado a la izquierda de un sub-bloque 4x4 superior a lo largo de un límite izquierdo del bloque actual, un segundo bloque vecino izquierdo que está situado a la izquierda de un sub-bloque 4x4 medio a lo largo del límite izquierdo del bloque actual y un tercer bloque vecino izquierdo que está situado a la izquierda de un sub-bloque 4x4 inferior a lo largo del límite izquierdo del bloque actual.

En algunos ejemplos, los circuitos de procesamiento del decodificador de vídeo 30 pueden usar hardware de comunicación (un receptor por cable o inalámbrico) del decodificador de vídeo 30 para recibir, en un flujo de bits de vídeo codificado, uno entre un conjunto de parámetros de secuencia (SPS), un conjunto de parámetros de imagen (PPS) o un encabezamiento de fragmento (SH). En estos ejemplos, los circuitos de procesamiento del decodificador de vídeo 30 pueden decodificar, a partir del SPS, el PPS o el SH recibido, una indicación del bloque vecino seleccionado para uno o más bloques en una unidad de datos de vídeo que corresponde al SPS, el PPS o el SH respectivo. En algunos ejemplos, los circuitos de procesamiento del decodificador de vídeo 30 pueden comparar una anchura del bloque actual con una altura del bloque actual para determinar una forma del bloque actual. En estos ejemplos, para seleccionar el bloque vecino, los circuitos de procesamiento del decodificador de vídeo 30 pueden seleccionar el bloque vecino entre uno de la pluralidad de bloques vecinos izquierdos o la pluralidad de bloques vecinos superiores basándose en la forma determinada del bloque actual.

En algunos ejemplos, los circuitos de procesamiento del decodificador de vídeo 30 pueden comparar al menos una entre una anchura del bloque actual o una altura del bloque actual, respectivamente, con al menos una entre una anchura mínima predeterminada para el bloque actual o una altura mínima predeterminada para el bloque actual, y puede determinar al menos uno entre un tipo de árbol para la división del bloque actual o una dirección de división para la división del bloque actual basándose en la comparación. En algunos ejemplos, el bloque actual es una unidad de codificación (CU) actual. En algunos de estos ejemplos, los circuitos de procesamiento del decodificador de vídeo 30 pueden determinar que una unidad de árbol de codificación (CTU) de los datos de vídeo abarca un límite de imagen de manera que un tamaño de la CTU se extiende más allá de una región de relleno de una imagen actual, y basándose en la CTU que abarca el límite de imagen, los circuitos de procesamiento del decodificador de vídeo 30 pueden dividir recursivamente la CTU usando un esquema de división predeterminado de la estructura de árboles de tipos múltiples para formar una pluralidad de unidades de codificación (CU) que incluye la CU actual, de manera que la CU actual está situada totalmente dentro de la imagen actual.

Se debe reconocer que, dependiendo del ejemplo, determinados actos o eventos de cualquiera de las técnicas descritas en esta solicitud se pueden realizar en una secuencia diferente, se pueden añadir, fusionar u omitir todos juntos (por ejemplo, no todos los actos o eventos descritos son necesarios para la práctica de las técnicas). Es más, en determinados ejemplos, los actos o eventos se pueden realizar simultáneamente, por ejemplo, a través de procesamiento de multihilo, procesamiento interrumpido o múltiples procesadores, en lugar de secuencialmente.

En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, firmware o cualquier combinación de estos. Si se implementan en software, las funciones pueden almacenarse o transmitirse como una o más instrucciones o código en un medio legible por ordenador y ejecutarse por una unidad de tratamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible tal como medios de almacenamiento de datos, o medios de comunicación que incluyen cualquier medio que facilite la transferencia de un programa informático de un lugar a otro, por ejemplo, según un protocolo de comunicación. De esta manera, los medios legibles por ordenador generalmente pueden corresponder a (1) un medio de almacenamiento tangible legible por ordenador que no sea transitorio o (2) un medio de

comunicación tal como una señal u onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que se pueda acceder mediante 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 descripción. Un producto de programa informático puede incluir un medio legible por ordenador.

A modo de ejemplo, y sin limitación, dichos medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento en disco óptico, almacenamiento en disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que se pueda utilizar para almacenar el código de programa deseado en forma de instrucciones o estructuras de datos y al que se pueda acceder mediante un ordenador. Además, cualquier conexión se denomina correctamente 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 abonado digital (DSL, Digital Subscriber Line) o tecnologías inalámbricas como infrarrojos, radio y microondas, a continuación, el cable coaxial, el cable de fibra óptica, el par trenzado, el DSL o las tecnologías inalámbricas como infrarrojos, radio y 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 en su lugar están dirigidos a medios de almacenamiento tangibles no transitorios. Disco, como se usa en esta invención, incluye disco compacto (Compact Disk, CD), disco láser, disco óptico, disco versátil digital (Digital Versatile Disk, DVD), disquete y disco Blu-ray, donde unos discos generalmente reproducen datos magnéticamente, mientras que otros discos reproducen datos ópticamente con láseres. Las combinaciones de lo anterior también deben incluirse dentro el alcance de los medios legibles por ordenador.

Las instrucciones pueden ser ejecutadas por uno o más procesadores, como uno o más procesadores de señales digitales (Digital Signal Procesadores, DSP), microprocesadores de propósito general, circuitos integrados específicos de aplicación (Application Specific Integrated Circuits, ASIC), matrices de puertas programables de campo (Field Programmable Gate Array, FPGA), circuitos de procesamiento de función fija, circuitos de procesamiento programables u otros circuitos lógicos discretos o integrados equivalentes. En consecuencia, el término "procesador", como se usa en esta invención, se puede referir a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en esta invención. Además, en algunos aspectos, la funcionalidad descrita en esta invención se puede proporcionar dentro de módulos de hardware y/o software dedicados configurados para codificar y decodificar, o incorporarse en un códec combinado. Además, las técnicas se podrían implementar completamente en uno o más circuitos o elementos lógicos.

Las técnicas de esta descripción se pueden implementar en una amplia variedad de dispositivos o aparatos, incluyendo un teléfono inalámbrico, un circuito integrado (CI) o un conjunto de CI (por ejemplo, un conjunto de chips). En esta descripción, se describen varios componentes, módulos o unidades para enfatizar los aspectos funcionales de los dispositivos configurados para realizar las técnicas descritas, pero no necesariamente requieren su realización a través de diferentes unidades de hardware. Más bien, como se ha descrito anteriormente, varios conjuntos se pueden combinar en una unidad de hardware de códec o proporcionar por medio de una colección de unidades de hardware interoperativas, que incluyen uno o más procesadores como se ha descrito anteriormente, junto con software y/o firmware adecuados.

Se han descrito varios ejemplos.

REIVINDICACIONES

1. Un procedimiento (100; 120) de codificación o decodificación de datos de vídeo, comprendiendo el procedimiento:

la comparación (102; 122) de al menos un valor de una dimensión de un bloque actual de los datos de vídeo con al menos un valor de una dimensión correspondiente de al menos un bloque vecino del bloque actual para obtener al menos un valor de dimensión relativa, estando el al menos un bloque vecino colocado adyacente al bloque actual;
basándose en al menos uno del al menos un valor de dimensión relativa, determinación de si el bloque actual debe ser dividido según una parte de árbol de predicción, PT, de un esquema de división basada en árboles de tipos múltiples; y,
cuando se determina que el bloque actual debe ser dividido según la parte de PT del esquema de división basada en árboles de tipos múltiples, basándose en al menos uno del al menos un valor de dimensión relativa, la determinación (104; 124) de si la división se hará según una estructura de árbol binario o una estructura de árbol ternario del lado central; y
cuando se determina que el bloque actual debe ser dividido según la parte de PT del esquema de división basada en árboles de tipos múltiples, división (106; 126) del bloque actual según la que se haya determinado de entre una estructura de árbol binario y una estructura de árbol ternario del lado central para formar una pluralidad de sub-bloques.

2. El procedimiento según la reivindicación 1,

donde el al menos un bloque vecino comprende un bloque vecino superior que está situado encima del bloque actual,
donde la comparación del valor de la dimensión del bloque actual con el valor de la dimensión correspondiente del bloque vecino superior comprende la comparación de una anchura del bloque actual con una anchura del bloque vecino superior, y
donde se determina dividir el bloque actual según una parte de PT de un esquema de división basada en árboles de tipos múltiples donde la anchura del bloque actual es mayor que la anchura del bloque vecino superior, y que la división del bloque actual comprende la división vertical del bloque actual.

3. El procedimiento según la reivindicación 1,

donde el al menos un bloque vecino comprende un bloque vecino izquierdo que está situado a la izquierda del bloque actual,
donde la comparación de la dimensión del bloque actual con la dimensión correspondiente del bloque vecino izquierdo comprende la comparación de una altura del bloque actual con una altura del bloque vecino izquierdo, y
donde se determina dividir el bloque actual según una parte de PT de un esquema de división basada en árboles de tipos múltiples donde la altura del bloque actual es mayor que la altura del bloque vecino izquierdo y que la división del bloque actual comprende la división horizontal del bloque actual.

4. El procedimiento según la reivindicación 1,

donde el bloque vecino es un bloque vecino en diagonal que comprende uno entre:

un bloque vecino superior izquierdo que está situado encima y a la izquierda del bloque actual,
un bloque vecino superior derecho que está situado encima y a la derecha del bloque actual, o
un bloque vecino inferior izquierdo que está situado debajo y a la derecha del bloque actual;

donde la comparación del valor de la dimensión del bloque actual con el valor de la dimensión correspondiente del bloque vecino en diagonal comprende la comparación de un área del bloque actual con un área del bloque vecino en diagonal; y
donde se determina dividir el bloque actual según una parte de PT de un esquema de división basada en árboles de tipos múltiples donde el área del bloque actual es mayor que el área del bloque vecino en diagonal.

5. El procedimiento según la reivindicación 1, comprendiendo además la selección del bloque vecino entre uno de:

- (i) una pluralidad de bloques vecinos superiores que incluye un primer bloque vecino superior que está situado encima de un sub-bloque 4x4 situado más a la izquierda a lo largo de un límite superior del bloque actual, un segundo bloque vecino superior que está situado encima de un sub-bloque 4x4 medio a lo largo del límite superior del bloque actual y un tercer bloque vecino superior que está situado encima de un sub-bloque 4x4 situado más a la derecha a lo largo del límite superior del bloque actual, o
- (ii) una pluralidad de bloques vecinos izquierdos que incluye un primer bloque vecino izquierdo que está situado a

la izquierda de un sub-bloque 4x4 superior a lo largo de un límite izquierdo del bloque actual, un segundo bloque vecino izquierdo que está situado a la izquierda de un sub-bloque 4x4 medio a lo largo del límite izquierdo del bloque actual y un tercer bloque vecino izquierdo que está situado a la izquierda de un sub-bloque 4x4 inferior a lo largo del límite izquierdo del bloque actual.

6. El procedimiento según la reivindicación 5, comprendiendo la decodificación de uno o más sub-bloques de la pluralidad de sub-bloques.

7. El procedimiento según la reivindicación 6, comprendiendo el procedimiento además:

la recepción, en un flujo de bits de vídeo codificado, de al menos uno entre un conjunto de parámetros de secuencia, SPS, un conjunto de parámetros de imagen, PPS, o un encabezamiento de fragmento, SH; y la decodificación, a partir del SPS, el PPS o el SH recibido, de una indicación del bloque vecino seleccionado para uno o más bloques en una unidad de datos de vídeo que corresponde al SPS, el PPS o el SH respectivo.

8. El procedimiento según la reivindicación 5, comprendiendo la codificación de los datos de vídeo, comprendiendo el procedimiento además:

la codificación de la pluralidad de sub-bloques; y la señalización de los datos de vídeo codificados que representan la pluralidad codificada de sub-bloques en un flujo de bits de vídeo codificado.

9. El procedimiento según la reivindicación 8, comprendiendo el procedimiento además: la señalización, en uno entre un conjunto de parámetros de secuencia, SPS, un conjunto de parámetros de imagen, PPS, o un encabezamiento de fragmento, SH, de una indicación del bloque vecino seleccionado para uno o más bloques en una unidad de datos de vídeo que corresponde al SPS, el PPS o el SH respectivo.

10. El procedimiento según la reivindicación 5, comprendiendo además:

la comparación de una anchura del bloque actual con una altura del bloque actual para determinar una forma del bloque actual, donde la selección del bloque vecino comprende la selección del bloque vecino entre uno de la pluralidad de bloques vecinos izquierdos o la pluralidad de bloques vecinos superiores basándose en la forma determinada del bloque actual.

11. El procedimiento según la reivindicación 1, comprendiendo la decodificación de datos de vídeo codificados, comprendiendo el procedimiento además:

la comparación de al menos una entre una anchura del bloque actual o una altura del bloque actual, respectivamente, con al menos una entre una anchura mínima predeterminada para el bloque actual o una altura mínima predeterminada para el bloque actual; y la determinación de al menos uno entre un tipo de árbol para la división del bloque actual o una dirección de división para la división del bloque actual basándose en la comparación.

12. El procedimiento según la reivindicación 1, donde el bloque actual es una unidad de codificación, CU, actual comprendiendo la decodificación de datos de vídeo codificados, comprendiendo el procedimiento además:

la determinación de que una unidad de árbol de codificación, CTU, de los datos de vídeo abarca un límite de imagen de manera que un tamaño de la CTU se extiende más allá de una región de relleno de una imagen actual; y basándose en la CTU que abarca el límite de imagen, la división recursiva de la CTU usando un esquema de división predeterminado de la estructura de árboles de tipos múltiples para formar una pluralidad de unidades de codificación, CU, que incluye la CU actual, donde la CU actual está situada totalmente dentro de la imagen actual.

13. El procedimiento según la reivindicación 1, comprendiendo la codificación de los datos de vídeo, comprendiendo el procedimiento además:

la asignación de una palabra de código respectiva de una pluralidad de palabras de código a una combinación de tipo de árbol de dirección respectiva de una pluralidad de combinaciones de tipos de árboles de dirección para la división del bloque actual,

donde la pluralidad de combinaciones de tipos de árboles de dirección incluye una combinación de árbol binario horizontal, una combinación de árbol binario vertical, una combinación de árbol ternario del lado central horizontal y una combinación de árbol ternario del lado central vertical, y

donde la división del bloque actual según la parte de PT del esquema de división basada en árboles de tipos múltiples comprende la división del bloque actual según una combinación de tipo de árbol de dirección particular incluida en la pluralidad de combinaciones de tipos de árboles de dirección; y

5 la señalización, en un flujo de bits de vídeo codificado, de la palabra de código respectiva asignada a la combinación de tipo de árbol de dirección particular según la cual se divide el bloque actual.

14. Un dispositivo para codificar o decodificar datos de vídeo, comprendiendo el dispositivo:

10 una memoria configurada para almacenar los datos de vídeo; y
circuitos de procesamiento en comunicación con la memoria, estando los circuitos de procesamiento configurados para:

15 comparar al menos un valor de una dimensión de un bloque actual de los datos de vídeo almacenados con al menos un valor de una dimensión correspondiente de al menos un bloque vecino del bloque actual para obtener al menos un valor de dimensión relativa, estando el al menos un bloque vecino colocado adyacente al bloque actual;
determinar, basándose en al menos uno del al menos un valor de dimensión relativa, si el bloque actual debe ser o no dividido según una parte de árbol de predicción, PT, de un esquema de división basada en árboles de tipos múltiples; y cuando se determina que el bloque actual debe ser dividido según la parte de PT del esquema de división basada en árboles de tipos múltiples, basándose en al menos uno del al menos un valor de dimensión relativa, la determinación de si se debe dividir según una estructura de árbol binario o una estructura de árbol ternario del lado central; y cuando se determina que el bloque actual debe ser dividido según la parte de PT del esquema de división basada en árboles de tipos múltiples, la división, basándose en la determinación, del bloque actual según la que se haya determinado entre una estructura de árbol binario y una estructura de árbol ternario del lado central para formar una pluralidad de sub-bloques.

20
25

15. El dispositivo según la reivindicación 14, donde los circuitos de procesamiento están configurados para realizar el procedimiento según cualquiera de las reivindicaciones 2 a 13.

30

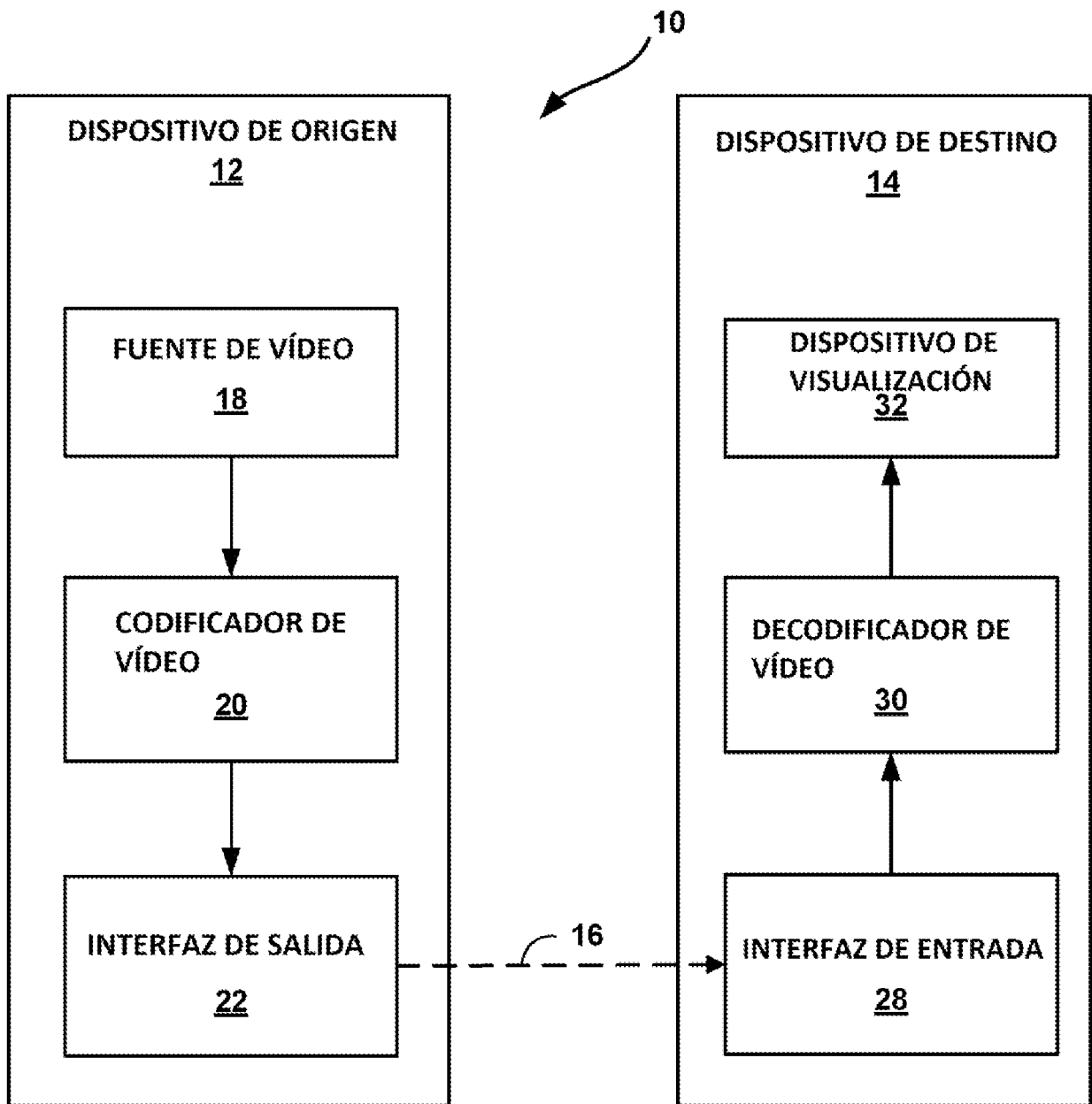


FIG. 1

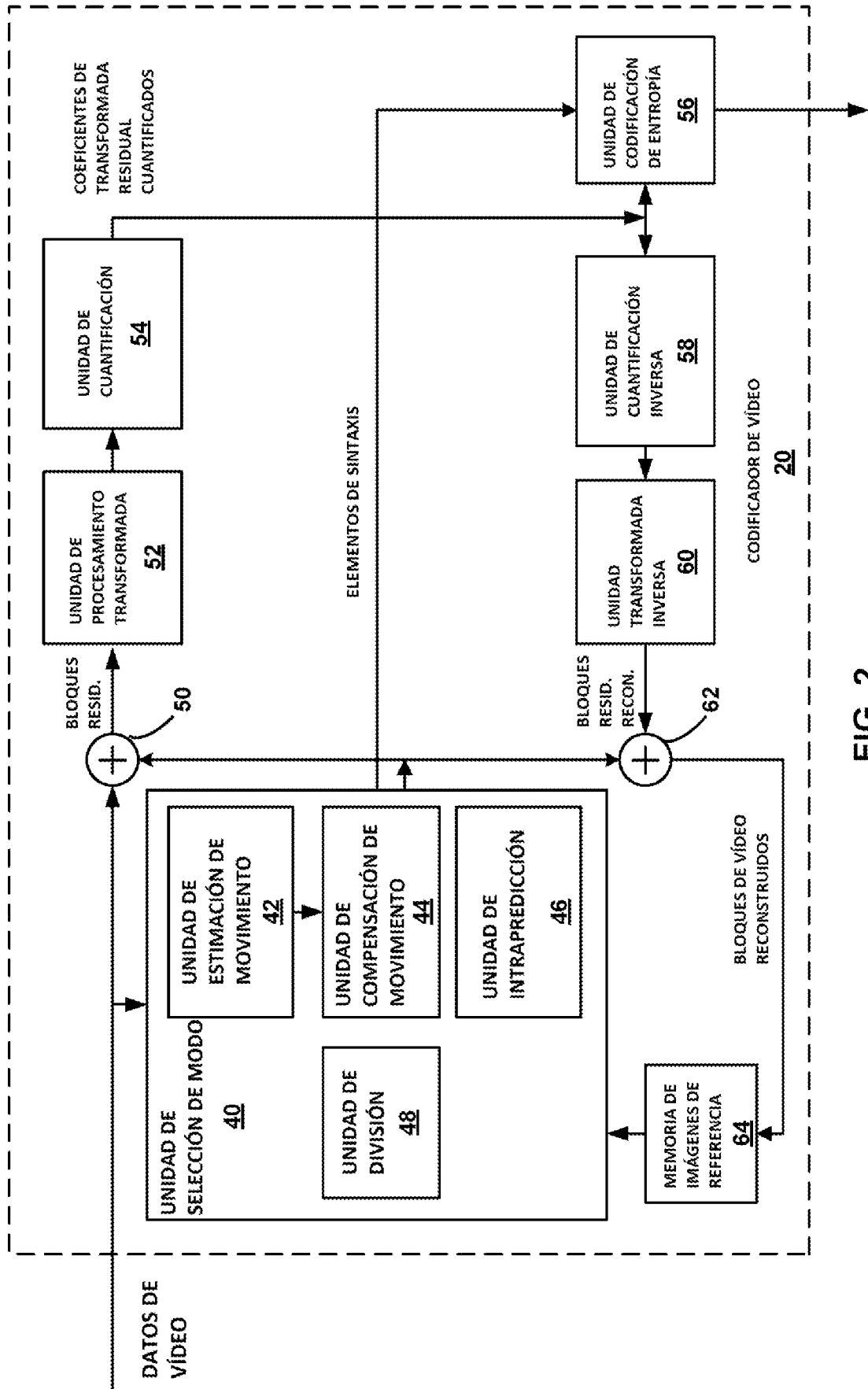


FIG. 2

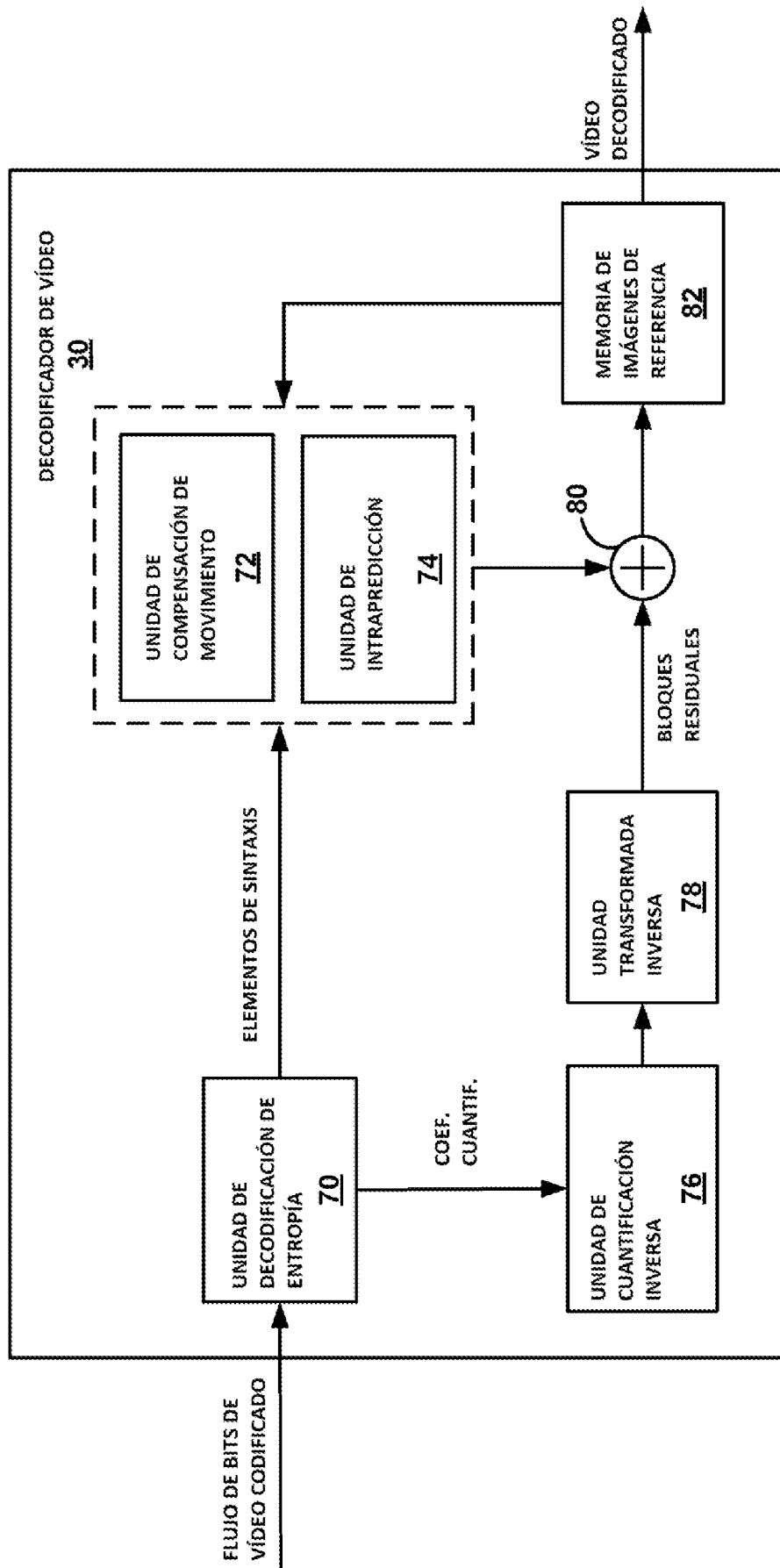


FIG. 3

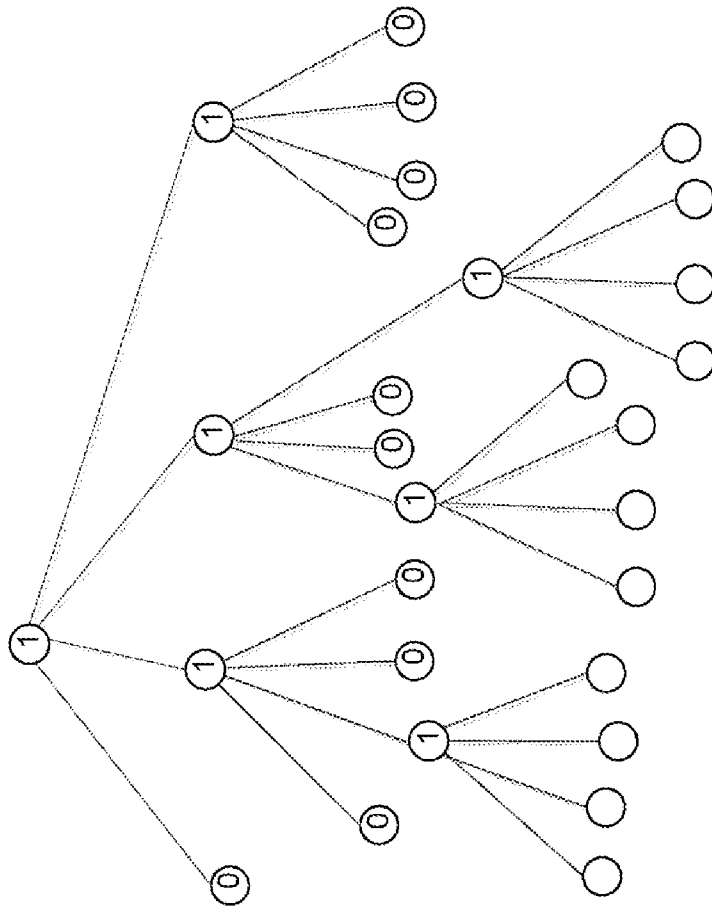


FIG. 4B

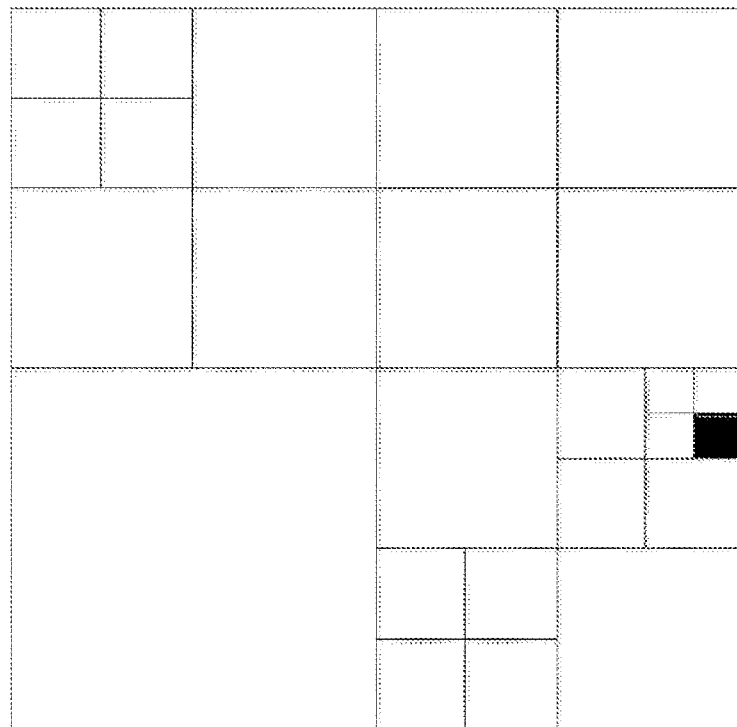


FIG. 4A

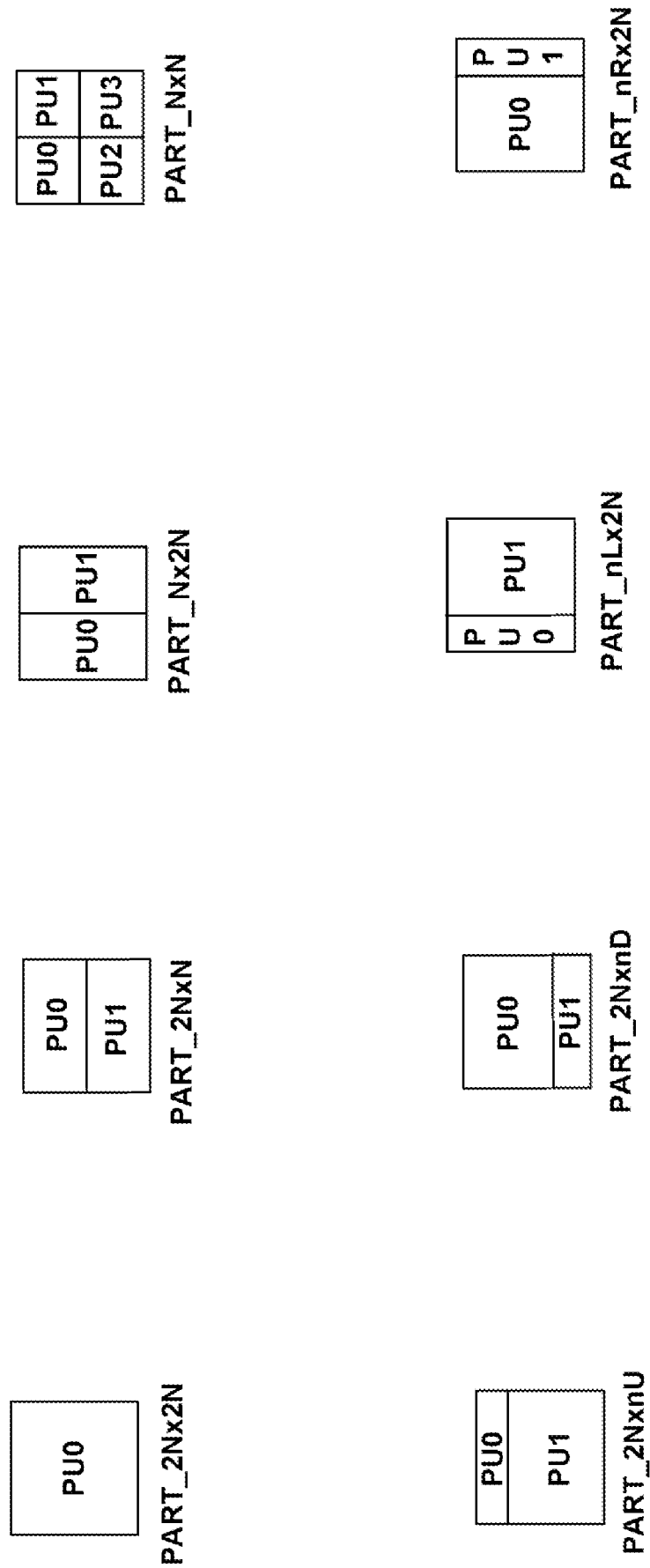


FIG. 5

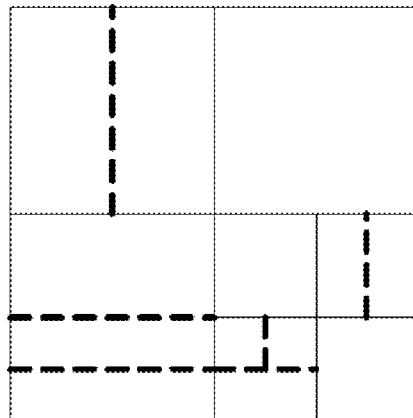


FIG. 6A

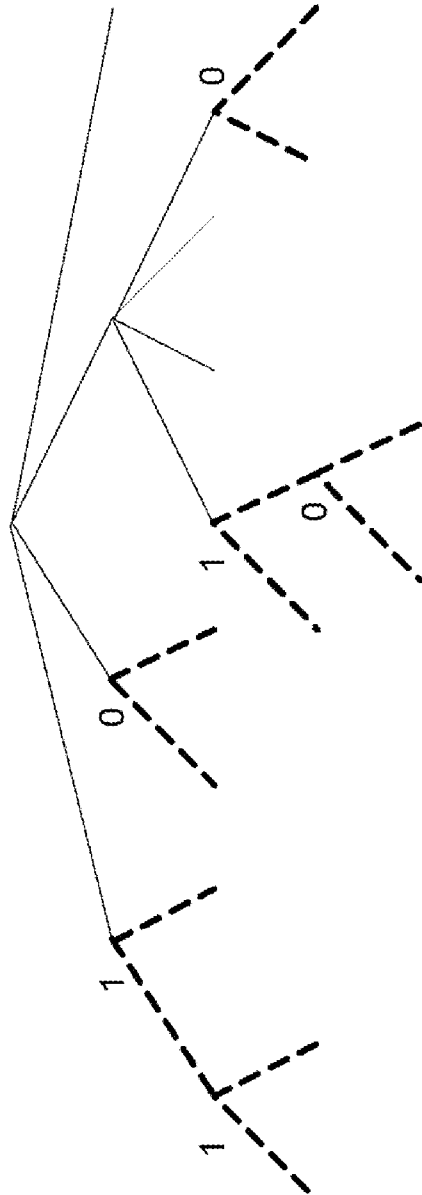


FIG. 6B

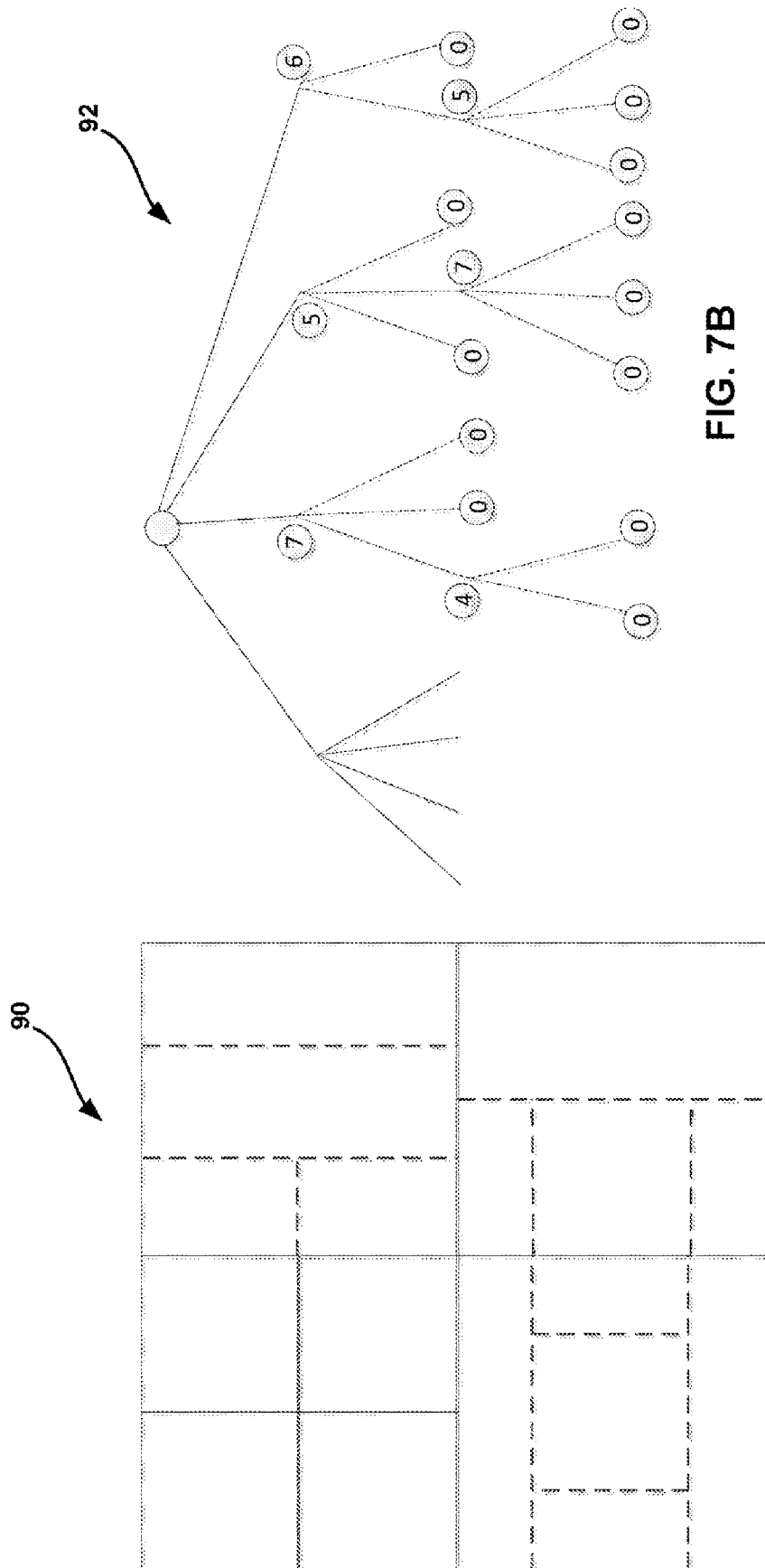


FIG. 7A

FIG. 7B

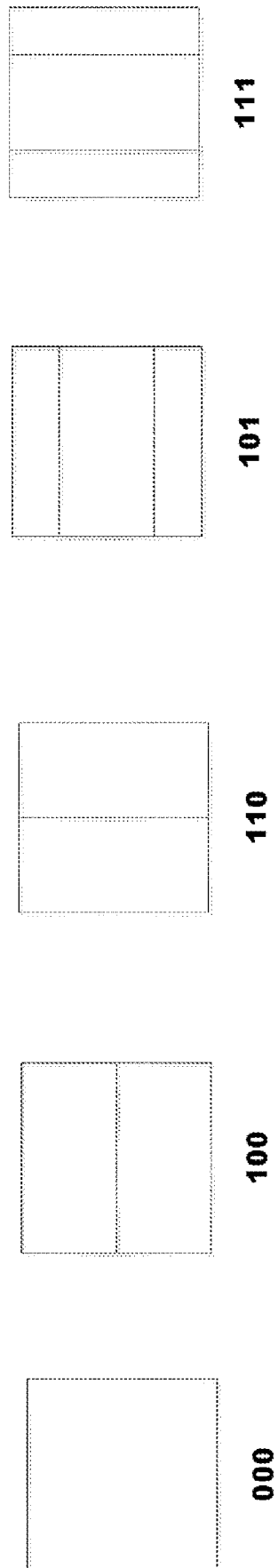


FIG. 8

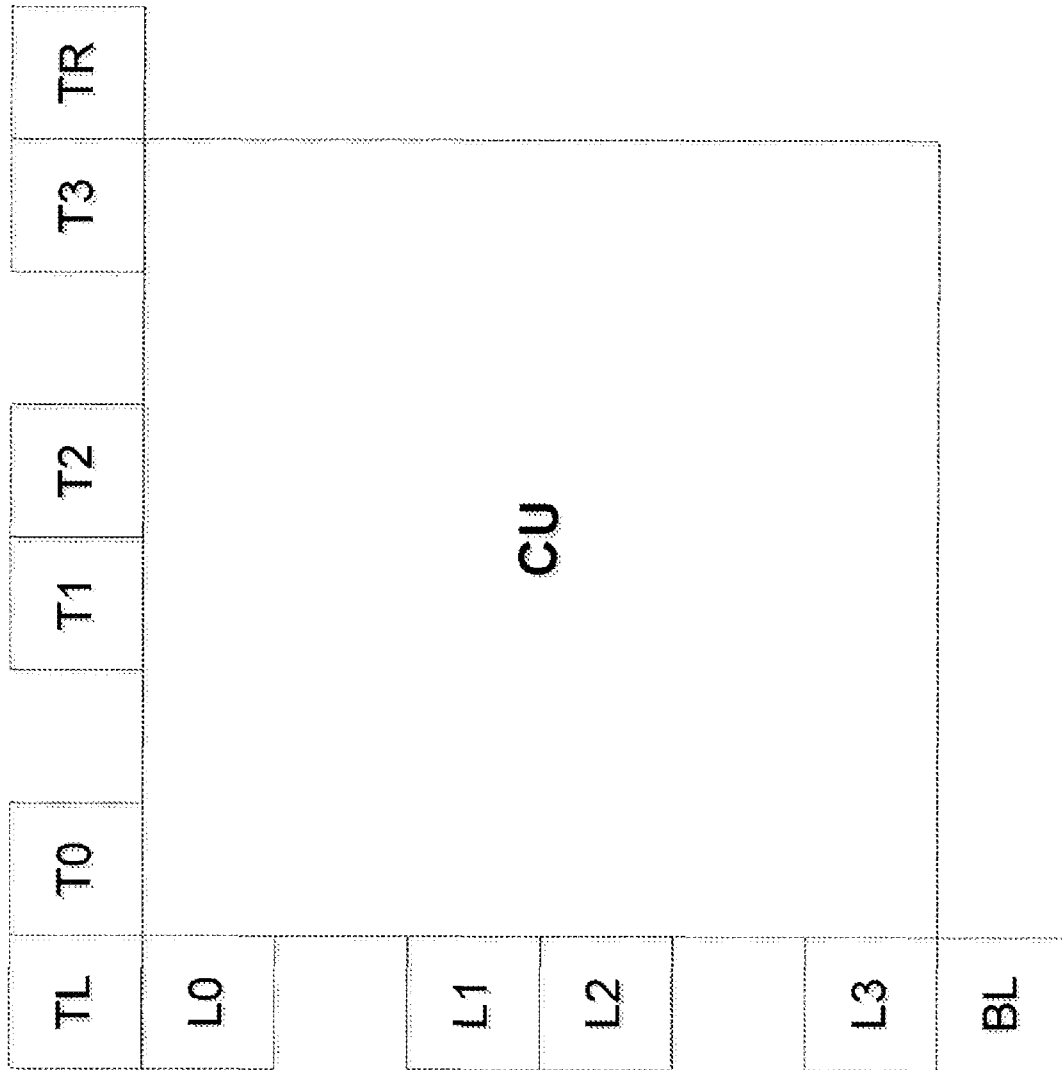


FIG. 9

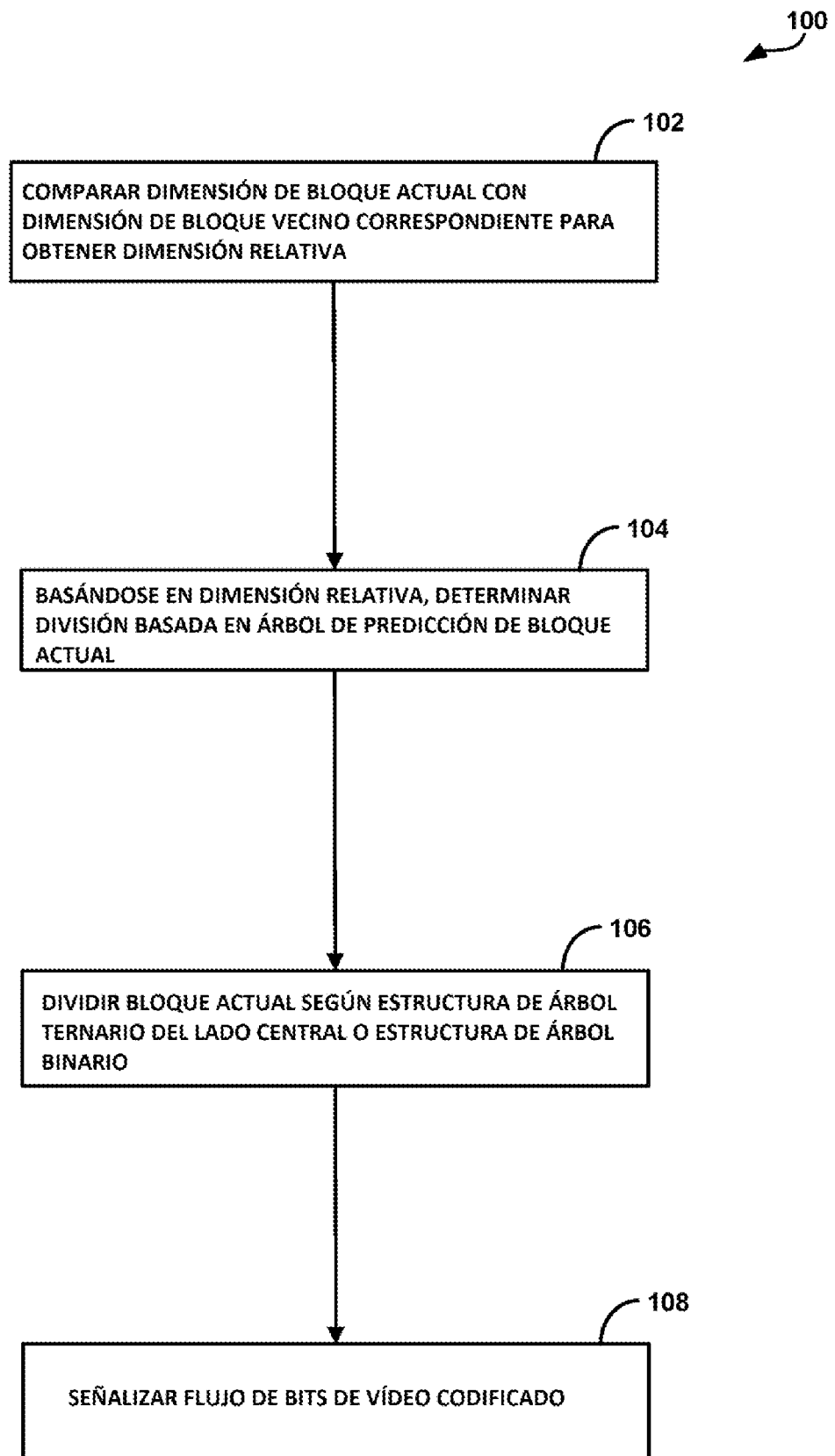


FIG. 10

120

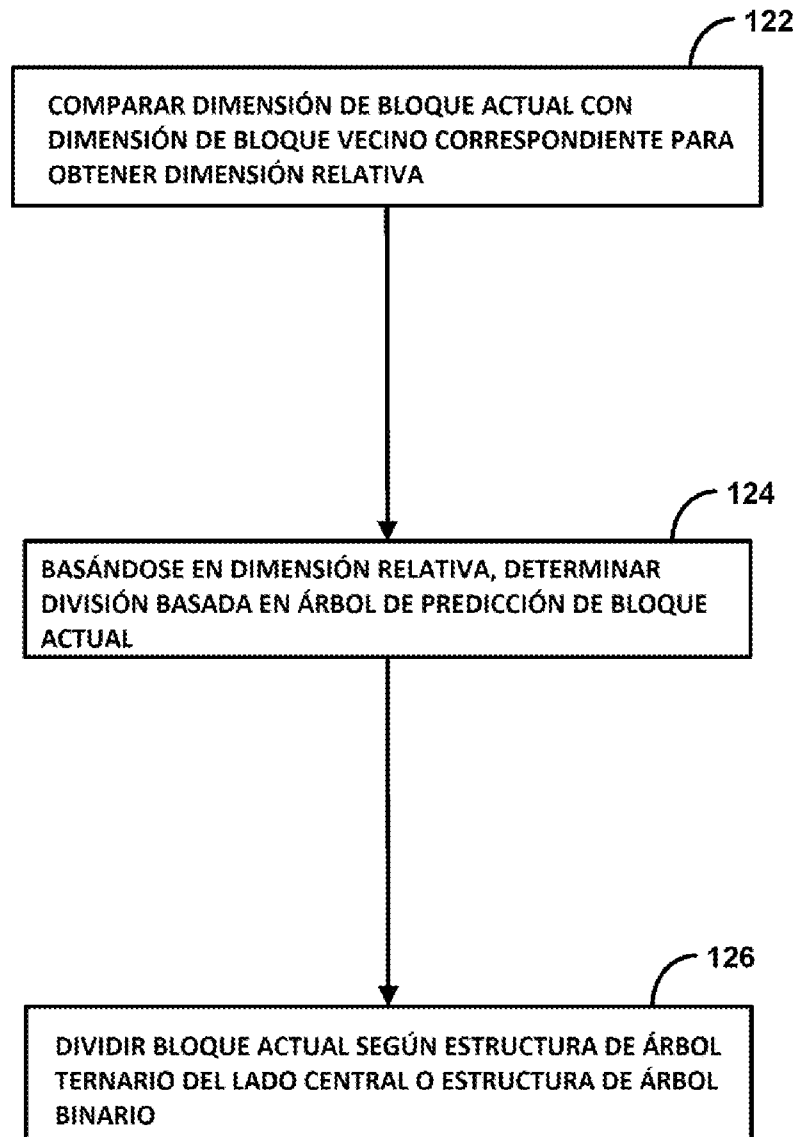


FIG. 11