

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 750 124**

51 Int. Cl.:

H04N 19/119 (2014.01)
H04N 19/18 (2014.01)
H04N 19/61 (2014.01)
H04N 19/70 (2014.01)
H04N 19/46 (2014.01)
H04N 19/129 (2014.01)
H04N 19/132 (2014.01)
H04N 19/14 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **29.06.2012 PCT/US2012/044990**
- 87 Fecha y número de publicación internacional: **03.01.2013 WO13003747**
- 96 Fecha de presentación y número de la solicitud europea: **29.06.2012 E 12733833 (3)**
- 97 Fecha y número de publicación de la concesión europea: **24.07.2019 EP 2727352**

54 Título: **Elementos sintácticos de señalización para coeficientes de transformada para subconjuntos de una unidad de codificación a nivel de hoja**

30 Prioridad:

30.06.2011 US 201161503541 P
27.10.2011 US 201161552341 P
06.03.2012 US 201213413475

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
25.03.2020

73 Titular/es:

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

72 Inventor/es:

KARCZEWICZ, MARTA;
WANG, XIANGLIN y
GUO, LIWEI

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 750 124 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

Elementos sintácticos de señalización para coeficientes de transformada para subconjuntos de una unidad de codificación a nivel de hoja

5

CAMPO TÉCNICO

[0001] Esta divulgación se refiere a la codificación y compresión de vídeo. Más específicamente, esta divulgación está dirigida a técnicas para escanear coeficientes de transformada cuantificados.

10

ANTECEDENTES

[0002] Las capacidades del vídeo digital pueden incorporarse a una amplia gama de dispositivos, que incluye televisores digitales, sistemas de difusión digital directa, sistemas de difusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, ordenadores de tableta, 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 denominados "teléfonos inteligentes", dispositivos de videoconferencia, dispositivos de transmisión de vídeo en tiempo real y similares. Los dispositivos de vídeo digital implementan técnicas de compresión de vídeo, tales como las descritas en las normas definidas por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, parte 10, codificación de vídeo avanzada (AVC), la norma de codificación de vídeo de alta eficacia (HEVC) actualmente en desarrollo y las ampliaciones de dichas normas. Los dispositivos de vídeo pueden transmitir, recibir, codificar, descodificar y/o almacenar información de vídeo digital con más eficacia implementando dichas técnicas de compresión de vídeo.

15

20

[0003] Las técnicas de compresión de vídeo realizan predicción espacial (intraimagen) y/o predicción temporal (interimagen) para reducir o eliminar la redundancia intrínseca 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) puede dividirse en bloques de vídeo, que también pueden denominarse bloques de árbol, unidades de codificación (CU) y/o nodos de codificación. Los bloques de vídeo en un fragmento intracodificado (I) de una imagen se codifican usando predicción espacial con respecto a muestras de referencia en bloques próximos en la misma imagen. Los bloques de vídeo en un fragmento intercodificado (P o B) de una imagen pueden usar la predicción espacial con respecto a muestras de referencia en bloques próximos en la misma imagen o la predicción temporal con respecto a muestras de referencia en otras imágenes de referencia. Las imágenes pueden denominarse tramas y las imágenes de referencia pueden denominarse tramas de referencia.

25

30

35

[0004] La predicción espacial o temporal da como resultado un bloque predictivo para un bloque a codificar. Los datos residuales representan diferencias de píxeles entre el bloque original a codificar y el bloque predictivo. Un bloque intercodificado se codifica de acuerdo con un vector de movimiento que apunta a un bloque de muestras de referencia que forman el bloque predictivo y los datos residuales que indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intracodificado se codifica de acuerdo con un modo de intracodificación y de acuerdo con los datos residuales. Para una mayor compresión, los datos residuales pueden transformarse desde el dominio de píxel a un dominio de transformada, dando como resultado coeficientes de transformada residuales, que a continuación se pueden cuantificar. Los coeficientes de transformada cuantificados, dispuestos inicialmente en una formación bidimensional, pueden escanearse con el fin de generar un vector unidimensional de coeficientes de transformada, y puede aplicarse codificación por entropía para lograr aún más compresión.

40

45

[0005] "A 4x4 CBP in H26L" de Louis Kerofsky, un documento presentado en la 12.^a reunión del Grupo de expertos en codificación de vídeo en Eibsee en enero de 2001, analiza una propuesta de un patrón de bloque codificado (CBP) que contiene información de qué bloques 8x8, luma y croma, contienen coeficientes de transformada, en los que un bloque 8x8 contiene 4 bloques 4x4, lo cual significa que la frase "el bloque 8x8 contiene coeficientes" significa que "uno o más de los 4 bloques 4x4 contienen coeficientes", pero no se proporciona información sobre cuál de los bloques 4x4 son omitidos. Los bloques 4x4 omitidos se codifican usando un símbolo EOB inicial, por lo que el símbolo EOB tiene dos propósitos: señalar el final de una serie de coeficientes codificados y señalar un bloque 4x4 omitido. Se asigna una palabra de código de 1 bit a EOB en ambos casos. De acuerdo con Kerofsky, este uso dual de EOB limita la libertad de diseñar un VLC para representar datos de coeficientes y, por lo tanto, Kerofsky propone un CBP 4x4 para señalar qué bloques 4x4 de un bloque 8x8 se omiten para que el VLC para los datos del coeficiente se pueda modificar sin alterar el bits utilizados para señalar bloques 4x4 omitidos. Kerofsky propone implementar esto a través de un CBP 4x4 sin cambiar el número de bits utilizados para la codificación eliminando los EOB iniciales y modificando la codificación del símbolo inicial en un bloque que no se omite.

50

55

60

[0006] En la presentación JCT-VC titulada "CE5: LCEC coded block flag coding under residual" de Chen et al (Equipo de Colaboración Conjunta sobre Codificación de Vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 5.^a reunión: Ginebra, Suiza, 16-23 de marzo de 2011, documento n.º JCTVC-E404, de fecha 20 de marzo de 2011), se analiza una propuesta para mejorar la eficiencia de codificación mediante la extensión de la codificación conjunta de indicadores de bloque codificados de componentes Y, U y V a nivel raíz CU. De acuerdo con la propuesta, la codificación conjunta incluye además un indicador de división de transformada del mismo bloque que el mismo nivel

65

si necesita señalizarse explícitamente, y la codificación conjunta no solo se realiza a nivel raíz CU, sino que también se extiende a un nivel de profundidad de transformada más alta para que se pueda maximizar el beneficio de dicha codificación conjunta.

5 **[0007]** En la presentación JCT-VC titulada "A unified design of RQT cbf coding in LCEC" de Li et al (Equipo Colaborativo Conjunto sobre Codificación de Vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 4.^a reunión: Daegu, Corea, 20-28 de enero de 2011, documento n.º JCTVC-D142, de fecha 24 de enero de 2011), se describe un esquema para mejorar la eficiencia de codificación de árbol cuádruple residual de LCEC (codificación por entropía de baja complejidad), en el que un indicador de bloque codificado se codifica solo si un bloque no se dividirá más.

SUMARIO

15 **[0008]** En la codificación de vídeo, para comprimir una cantidad de datos utilizados para representar datos de vídeo, un codificador de vídeo puede codificar por entropía los datos de vídeo. De acuerdo con las técnicas descritas en el presente documento, como parte de la codificación por entropía, un codificador de vídeo divide una unidad de datos de vídeo a nivel de hoja en una pluralidad de subconjuntos de coeficientes de transformada. Una unidad a nivel de hoja como se describe en el presente documento se refiere a una unidad no dividida de la estructura de datos de vídeo, un ejemplo de la cual es un nodo hijo final, no dividido de una estructura de datos de árbol cuádruple, como se describe con más detalle a continuación.

[0009] La invención se define mediante las reivindicaciones independientes.

25 **[0010]** Para al menos uno de los subconjuntos, el codificador genera un elemento sintáctico que indica si el subconjunto incluye algún coeficiente distinto de cero como parte de un flujo de bits codificado por entropía. El codificador determina si no se debe señalar el elemento sintáctico para un subconjunto de la pluralidad de subconjuntos de coeficientes de transformada. Por ejemplo, el codificador puede determinar si se debe señalar o no el elemento sintáctico basándose de una serie de coeficientes potenciales distintos de cero dentro del subconjunto, o basándose en un número medio de coeficientes distintos de cero para el subconjunto basándose en estadísticas para una o más unidades de datos de vídeo a nivel de hoja previamente codificadas.

35 **[0011]** Un descodificador puede leer un flujo de bits codificado por entropía y determinar si descodifica o no los coeficientes de transformada del subconjunto basándose en el elemento sintáctico. En algunos ejemplos, el descodificador puede determinar si descodifica o no un subconjunto de coeficientes de transformada basándose en si el flujo de bits codificado por entropía incluye o no un elemento sintáctico asociado con el subconjunto. Por ejemplo, si el subconjunto no incluye un elemento sintáctico asociado, el descodificador descodifica el subconjunto. Sin embargo, si el subconjunto incluye un elemento sintáctico asociado, el descodificador determina si descodifica o no el subconjunto basándose en un valor del elemento sintáctico. Por ejemplo, si el elemento sintáctico tiene un primer valor, el descodificador descodifica el subconjunto, pero si el elemento sintáctico tiene un segundo valor diferente, el descodificador no descodifica el subconjunto.

45 **[0012]** En algunos ejemplos, las técnicas de esta divulgación pueden mejorar la eficiencia de codificación de un codificador o descodificador. Por ejemplo, las técnicas descritas en el presente documento pueden reducir una cantidad de bits utilizados por un codificador para generar un flujo de bits codificado por entropía que representa datos de vídeo.

50 **[0013]** En un ejemplo, un procedimiento para codificar una unidad de datos de vídeo comprende dividir una unidad a nivel de hoja de datos de vídeo en una pluralidad de subconjuntos de coeficientes de transformada y generar, para un subconjunto de la pluralidad de subconjuntos de coeficientes de transformada, un elemento sintáctico que indica si el subconjunto incluye o no coeficientes distintos de cero.

55 **[0014]** En otro ejemplo, un dispositivo puede configurarse para codificar al menos una unidad de datos de vídeo a nivel de hoja. El dispositivo puede comprender un procesador configurado para dividir una unidad de datos de vídeo a nivel de hoja en una pluralidad de subconjuntos de coeficientes de transformada, y generar, para un subconjunto de la pluralidad de subconjuntos de coeficientes de transformada, un elemento sintáctico que indica si el subconjunto incluye o no coeficientes distintos de cero.

60 **[0015]** En otro ejemplo, un dispositivo puede configurarse para codificar al menos una unidad de datos de vídeo a nivel de hoja, comprendiendo el dispositivo medios para dividir una unidad de datos de vídeo a nivel de hoja en una pluralidad de subconjuntos de coeficientes de transformada, y medios para generar, para un subconjunto de la pluralidad de subconjuntos de coeficientes de transformada, un elemento sintáctico que indica si el subconjunto incluye o no coeficientes distintos de cero.

65 **[0016]** En otro ejemplo, un procedimiento para descodificar una unidad de datos de vídeo comprende dividir una unidad a nivel de hoja de datos de vídeo en una pluralidad de subconjuntos de coeficientes de transformada, determinando, para un subconjunto de la pluralidad de subconjuntos de coeficientes de transformada, si el subconjunto

incluye un elemento sintáctico asociado que indica si el subconjunto incluye o no coeficientes distintos de cero, determinando, basándose en el elemento sintáctico, si descodifica o no el subconjunto.

[0017] En otro ejemplo, un dispositivo puede configurarse para descodificar una unidad de datos de vídeo, comprendiendo el dispositivo un procesador configurado para dividir una unidad de datos de vídeo a nivel de hoja en una pluralidad de subconjuntos de coeficientes de transformada, determinar, para un subconjunto de la pluralidad de subconjuntos de coeficientes de transformada, si el subconjunto incluye un elemento sintáctico asociado que indica si el subconjunto incluye o no coeficientes distintos de cero y determina, basándose en el elemento sintáctico, si descodifica o no el subconjunto.

[0018] En otro ejemplo, un dispositivo puede configurarse para descodificar una unidad de datos de vídeo, con el dispositivo que comprende medios para dividir una unidad de datos de vídeo a nivel de hoja en una pluralidad de subconjuntos de coeficientes de transformada, medios para determinar un subconjunto de la pluralidad de subconjuntos de coeficientes de transformada, si el subconjunto incluye un elemento sintáctico que indica si el subconjunto incluye o no coeficientes distintos de cero, y medios para determinar, basándose en el elemento sintáctico, si descodificar o no el subconjunto.

[0019] Las técnicas descritas en esta divulgación pueden implementarse en hardware, software, firmware o en cualquier combinación de los mismos. Por ejemplo, diversas técnicas pueden implementarse o ejecutarse por uno o más procesadores. Como se usa en el presente documento, un procesador puede referirse a un microprocesador, a un circuito integrado específico de la aplicación (ASIC), a una matriz de puerta programable por campo (FPGA), a un procesador de señales digitales (DSP), o a otro circuito lógico integrado o discreto equivalente. El software puede ejecutarse por uno o más procesadores. El software que comprende instrucciones para ejecutar las técnicas se puede almacenar inicialmente en un medio legible por ordenador y cargarse y ejecutarse mediante un procesador.

[0020] Por consiguiente, esta divulgación también contempla medios de almacenamiento legibles por ordenador que comprenden instrucciones para hacer que un procesador (u otro dispositivo informático) realice cualquiera de una variedad de técnicas descritas en esta divulgación. En algunos casos, el medio de almacenamiento legible por ordenador puede formar parte de un producto de almacenamiento de programa informático, que puede venderse a los fabricantes y/o usarse en un dispositivo. El producto de programa informático puede incluir el medio legible por ordenador y, en algunos casos, también puede incluir materiales de embalaje.

[0021] En un ejemplo, esta divulgación describe un medio de almacenamiento legible por ordenador que almacena instrucciones que, cuando se ejecutan, hacen que un dispositivo informático divida una unidad a nivel de hoja de datos de vídeo en una pluralidad de subconjuntos de coeficientes de transformada y genere, para un subconjunto de la pluralidad de subconjuntos de coeficientes de transformada, un elemento sintáctico que indica si el subconjunto de incluye algún coeficientes distintos de cero.

[0022] En otro ejemplo, esta divulgación describe un medio de almacenamiento legible por ordenador que almacena instrucciones que, cuando se ejecutan, hacen que un dispositivo informático divida una unidad de datos de vídeo a nivel de hoja en una pluralidad de subconjuntos de coeficientes de transformada, determine, un subconjunto de la pluralidad de subconjuntos de coeficientes de transformada, si el subconjunto incluye un elemento sintáctico asociado que indica si el subconjunto incluye o no coeficientes distintos de cero y determine, basándose en el elemento sintáctico, si descodifica el subconjunto.

[0023] Los detalles de uno o más ejemplos se exponen en los dibujos adjuntos y en la siguiente descripción. Otras características, objetivos y ventajas de la invención resultarán evidentes a partir de la descripción y de los dibujos y a partir de las reivindicaciones.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

[0024]

La FIG. 1 es un diagrama de bloques que ilustra un ejemplo de sistema de codificación y descodificación de vídeo configurado para funcionar de acuerdo con las técnicas de esta divulgación.

La FIG. 2 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo configurado para funcionar de acuerdo con las técnicas de esta divulgación.

La FIG. 3 es un diagrama de bloques que ilustra un ejemplo de un descodificador de vídeo configurado para funcionar de acuerdo con las técnicas de esta divulgación.

La FIG. 4 es un diagrama conceptual que ilustra un ejemplo de una unidad de datos de vídeo a nivel de hoja dividida en una pluralidad de subconjuntos de coeficientes de transformada consistentes con uno o más aspectos de esta divulgación.

La FIG. 5 es un diagrama de flujo que ilustra un ejemplo de un procedimiento de codificación de una unidad de datos de vídeo a nivel de hoja consistente con uno o más aspectos de esta divulgación.

5 La FIG. 6 es un diagrama de flujo que ilustra otro ejemplo de un procedimiento de codificación de una unidad de datos de vídeo a nivel de hoja consistente con uno o más aspectos de esta divulgación.

La FIG. 7 es un diagrama de flujo que ilustra otro ejemplo de un procedimiento de codificación de una unidad de datos de vídeo a nivel de hoja consistente con uno o más aspectos de esta divulgación.

10 La FIG. 8 es un diagrama de flujo que ilustra un ejemplo de un procedimiento de descodificación de una unidad de datos de vídeo a nivel de hoja consistente con uno o más aspectos de esta divulgación.

DESCRIPCIÓN DETALLADA

15 **[0025]** La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y descodificación de vídeo 10 de ejemplo que puede utilizar las técnicas descritas en esta divulgación. Como se muestra en la FIG. 1, el sistema 10 incluye un dispositivo de origen 12 que genera datos de vídeo codificados que un dispositivo de destino 14 va a descodificar en un momento posterior. El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender cualquiera entre una amplia gama de dispositivos, incluidos ordenadores de escritorio, ordenadores plegables (es decir, portátiles), ordenadores de tableta, descodificadores, equipos telefónicos de mano tales como los denominados teléfonos "inteligentes", los denominados paneles "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de transmisión continua 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.

25 **[0026]** El dispositivo de destino 14 puede recibir los datos de vídeo codificados que se van a descodificar a través de un enlace 16. El enlace 16 puede comprender cualquier tipo de medio o dispositivo capaz de transferir los datos de vídeo codificados desde el dispositivo de origen 12 hasta el dispositivo de destino 14. En un ejemplo, el enlace 16 puede comprender un medio de comunicación para permitir que el dispositivo de origen 12 transmita datos de vídeo codificados directamente a un dispositivo de destino 14 en tiempo real. Los datos de vídeo codificados se pueden modular de acuerdo con 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, inalámbrica o cableada, tal como un espectro de radiofrecuencia (RF) o una o más líneas físicas de transmisión. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área amplia o una red global tal como Internet. El medio de comunicación puede incluir encaminadores, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo de origen 12 hasta el dispositivo de destino 14.

40 **[0027]** De forma alternativa, los datos codificados pueden enviarse desde la interfaz de salida 22 hasta un dispositivo de almacenamiento 32. De manera similar, se puede acceder a los datos codificados desde el dispositivo de almacenamiento 32 mediante una interfaz de entrada 28. El dispositivo de almacenamiento 32 puede incluir cualquiera de una variedad de medios de almacenamiento de datos de acceso distribuido o local, tales como una unidad de disco duro, unos discos Blu-ray, unos DVD, unos CD-ROM, una memoria flash, una memoria volátil o no volátil o cualquier otro medio de almacenamiento digital adecuado para almacenar datos de vídeo codificados. En otro ejemplo, el dispositivo de almacenamiento 32 puede corresponder a un servidor de archivos o a otro dispositivo de almacenamiento intermedio que puede contener el vídeo codificado generado por el dispositivo de origen 12. El dispositivo de destino 14 puede acceder a datos de vídeo almacenados en el dispositivo de almacenamiento 32 a través de transmisión en tiempo real 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 a modo de ejemplo de ficheros incluyen un servidor de la Red (por ejemplo, para una sede de la Red), un servidor del FTP, dispositivos de almacenamiento conectados en red (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 estándar, incluida una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión de Wi-Fi), una conexión cableada (por ejemplo, DSL, módem de cable, etc.) o una combinación de ambas que sea adecuada para acceder a datos de vídeo codificados almacenados en un servidor de ficheros. La transmisión de datos de vídeo codificados desde el dispositivo de almacenamiento 32 puede ser una transmisión continua, una transmisión de descarga o una combinación de ambas.

60 **[0028]** Las técnicas de esta divulgación no están limitadas necesariamente a aplicaciones o configuraciones inalámbricas. Las técnicas se pueden aplicar a la codificación de vídeo como apoyo a cualquiera de una variedad de aplicaciones multimedia, tales como radiodifusiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones continuas de vídeo, por ejemplo, a través de Internet, codificación de vídeo digital para su almacenamiento en un medio de almacenamiento de datos, descodificación de vídeo digital almacenado en un medio de almacenamiento de datos, u otras aplicaciones. En algunos ejemplos, el sistema 10 puede estar configurado para prestar soporte a una transmisión de vídeo unidireccional o bidireccional, a

fin de prestar soporte a aplicaciones tales como la transmisión continua de vídeo, la reproducción de vídeo, la radiodifusión de vídeo y/o la videotelefonía.

5 **[0029]** En el ejemplo de la FIG. 1, el dispositivo de origen 12 incluye un origen de vídeo 18, un codificador de vídeo 20 y una interfaz de salida 22. En algunos casos, la interfaz de salida 22 puede incluir un modulador/desmodulador (módem) y/o un transmisor. En el dispositivo de origen 12, el origen de vídeo 18 puede incluir un origen tal como un dispositivo de captación de vídeo, por ejemplo, una videocámara, un archivo de vídeo que contiene vídeo captado previamente, una interfaz de entrada de vídeo para recibir vídeo desde un proveedor de contenido de vídeo y/o un sistema de gráficos por ordenador para generar datos de gráficos por ordenador como el vídeo de origen, o una combinación de dichos orígenes. Como un ejemplo, si el origen de vídeo 18 es una videocámara, el dispositivo de origen 12 y el dispositivo de destino 14 pueden formar los denominados teléfonos con cámara o videoteléfonos. Sin embargo, las técnicas descritas en esta divulgación pueden ser aplicables a la codificación de vídeo en general, y se pueden aplicar a aplicaciones inalámbricas y/o cableadas.

15 **[0030]** El codificador de vídeo 20 puede codificar el vídeo captado, captado previamente o generado por ordenador. Los datos de vídeo codificados se pueden transmitir directamente al dispositivo de destino 14 por medio de la interfaz de salida 22 del dispositivo de origen 12. Los datos de vídeo codificados se pueden almacenar de forma adicional (o alternativa) en el dispositivo de almacenamiento 32 para un posterior acceso por el dispositivo de destino 14 u otros dispositivos, para su descodificación y/o reproducción.

20 **[0031]** El dispositivo de destino 14 incluye una interfaz de entrada 28, un descodificador de vídeo 30 y un dispositivo de visualización 32. En algunos casos, la interfaz de entrada 28 puede incluir un receptor y/o un módem. La interfaz de entrada 28 del dispositivo de destino 14 recibe los datos de vídeo codificados a través del enlace 16. Los datos de vídeo codificados comunicados a través del enlace 16, o proporcionados en el dispositivo de almacenamiento 32, pueden incluir una diversidad de elementos sintácticos generados por el codificador de vídeo 20 para su uso por un descodificador de vídeo, tal como el descodificador de vídeo 30, en la descodificación de los datos de vídeo. Dichos elementos sintácticos se pueden incluir con los datos de vídeo codificados, transmitidos en un medio de comunicación, almacenados en un medio de almacenamiento o almacenados en un servidor de archivos.

30 **[0032]** El dispositivo de visualización 32 puede estar integrado en, o ser externo a, el dispositivo de destino 14. En algunos ejemplos, el dispositivo de destino 14 puede incluir un dispositivo de visualización integrado y también estar configurado para interconectarse con un dispositivo de visualización externo. En otros ejemplos, el dispositivo de destino 14 puede ser un dispositivo de visualización. En general, el dispositivo de visualización 32 muestra los datos de vídeo descodificados a un usuario, y puede comprender cualquiera de una variedad de dispositivos de visualización, tales como una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de visualización.

35 **[0033]** El codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con una norma de compresión de vídeo, tal como la norma de codificación de vídeo de alta eficiencia (HEVC), actualmente en fase de elaboración, y pueden ajustarse al modelo de prueba de HEVC (HM). De forma alternativa, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con otras normas de propiedad o industriales, tales como la norma ITU-T H.264, denominada de forma alternativa MPEG-4, parte 10, codificación avanzada de vídeo (AVC), o ampliaciones de dichas normas. Sin embargo, las técnicas de esta divulgación no están limitadas a ninguna norma de codificación particular. Otros ejemplos de normas de compresión de vídeo incluyen MPEG-2 e ITU-T H.263.

45 **[0034]** Aunque no se muestra en la FIG. 1, en algunos aspectos, tanto el codificador de vídeo 20 como el descodificador de vídeo 30 pueden estar integrados con un codificador y descodificador de audio, y pueden incluir unidades MUX-DEMUX adecuadas, u otro tipo de hardware y software, para gestionar la codificación tanto de audio como de vídeo en un flujo de datos común o en flujos de datos independientes. Si procede, en algunos ejemplos, las unidades MUX-DEMUX pueden ajustarse al protocolo de multiplexador ITU H.223 o a otros protocolos, tales como el protocolo de datagramas de usuario (UDP).

50 **[0035]** Tanto el codificador de vídeo 20 como el descodificador de vídeo 30 pueden implementarse como cualquiera de entre una variedad de circuitos codificadores adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), formaciones de puertas programables in situ (FPGA), lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. Cuando las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio adecuado, no transitorio, legible por ordenador, y ejecutar las instrucciones en hardware usando uno o más procesadores para realizar las técnicas de esta divulgación. Tanto el codificador de vídeo 20 como el descodificador de vídeo 30 se pueden incluir en uno o más codificadores o descodificadores, cualquiera de los cuales se puede integrar como parte de un codificador/descodificador (CÓDEC) combinado en un dispositivo respectivo.

60 **[0036]** El equipo JCT-VC está trabajando en el desarrollo de la norma HEVC. La iniciativa de normalización de la HEVC se basa en un modelo en evolución de un dispositivo de codificación de vídeo, denominado modelo de prueba de la HEVC (HM). El HM supone varias capacidades adicionales de los dispositivos de codificación de vídeo respecto a los dispositivos existentes, de acuerdo, por ejemplo, con la norma ITU-T H.264/AVC. Por ejemplo, mientras que la

norma H.264 proporciona nueve modalidades de codificación mediante predicción intra, el HM puede proporcionar hasta treinta y tres modalidades de codificación mediante predicción intra.

[0037] En general, el modelo de explotación del HM describe que una trama o imagen de vídeo puede dividirse en una secuencia de bloques de árbol o unidades de codificación más grandes (LCU), que incluyen tanto muestras de luminancia como de crominancia. Un bloque arbolado tiene un fin similar al de un macrobloque de la norma H.264. Un fragmento incluye un número de bloques arbolados consecutivos en orden de codificación. Una trama o imagen de vídeo puede dividirse en uno o más fragmentos. Cada bloque arbolado puede dividirse en unidades de codificación (CU) de acuerdo con un árbol cuádruple. Por ejemplo, un bloque arbolado, como un nodo raíz del árbol cuádruple, puede dividirse en cuatro nodos hijos, y cada nodo hijo puede, a su vez, ser un nodo padre y dividirse en otros cuatro nodos hijos. Un nodo hijo final, no dividido, como un nodo hoja del árbol cuádruple, comprende un nodo de codificación, es decir, un bloque de vídeo codificado. Tal nodo secundario final no dividido de una estructura de datos de vídeo se denomina en el presente documento una unidad a nivel de hoja. Los datos sintácticos asociados con un flujo de bits codificado pueden definir un número máximo de veces que puede dividirse un bloque arbolado, y también pueden definir un tamaño mínimo de los nodos de codificación.

[0038] Una CU incluye un nodo de codificación y unidades de predicción (PU) y unidades de transformada (TU) asociadas con el nodo de codificación. Un tamaño de la CU corresponde a un tamaño del nodo de codificación y debe ser de forma cuadrada. El tamaño de la CU puede variar desde 8x8 píxeles hasta el tamaño del bloque arbolado, con un máximo de 64x64 píxeles o más. Cada CU puede contener una o más PU y una o más TU. Los datos sintácticos asociados con una CU pueden describir, por ejemplo, la división de la CU en una o más PU. Las modalidades de división pueden diferir entre sí si la CU está codificada en modalidad de omisión o directa, codificada en modalidad de predicción intra o codificada en modalidad de predicción inter. Las PU pueden dividirse para tener forma no cuadrada. Los datos sintácticos asociados con una CU también pueden describir, por ejemplo, la división de la CU en una o más TU de acuerdo con un árbol cuádruple. Una TU puede tener forma cuadrada o no cuadrada.

[0039] La norma HEVC admite transformaciones de acuerdo con las TU, que pueden ser diferentes para diferentes CU. El tamaño de las TU se basa típicamente en el tamaño de las PU dentro de una CU dada definida para una LCU dividida, aunque puede que no sea siempre así. Las TU son típicamente del mismo tamaño o de un tamaño más pequeño que las PU. En algunos ejemplos, las muestras residuales correspondientes a una CU pueden subdividirse en unidades más pequeñas utilizando una estructura de árbol cuádruple conocida como "árbol cuádruple residual" (RQT). Los nodos hoja del RQT pueden denominarse unidades de transformada (TU). Tal nodo de hoja TU es un ejemplo de una unidad a nivel de hoja como se describe en el presente documento. Los valores de diferencias de píxeles asociados a las TU pueden transformarse para generar coeficientes de transformada, que pueden cuantificarse.

[0040] En general, una PU incluye datos relacionados con el proceso de predicción. Por ejemplo, cuando la PU se codifica mediante intramodo, la PU puede incluir datos que describen un modo de predicción intra para la PU. Como otro ejemplo, cuando la PU se codifica mediante intermodo, la PU puede incluir datos que definen un vector de movimiento para la PU. Los datos que definen el vector de movimiento para una PU pueden describir, por ejemplo, una componente horizontal del vector de movimiento, una componente vertical del vector de movimiento, una resolución para el vector de movimiento (por ejemplo, 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 (por ejemplo, lista 0, lista 1 o lista C) para el vector de movimiento.

[0041] En general, se usa una TU para los procesos de transformada y cuantificación. Una CU dada que tiene una o más PU también puede incluir una o más unidades de transformada (TU). Tras la predicción, el codificador de vídeo puede calcular valores residuales correspondientes a la PU. Los valores residuales comprenden valores de diferencias de píxeles que pueden transformarse en coeficientes de transformada, cuantificarse y escanearse usando las TU para generar coeficientes de transformada serializados para la codificación por entropía. Esta divulgación usa típicamente el término "bloque de vídeo" para referirse a un nodo de codificación de una CU. En algunos casos específicos, esta divulgación también puede usar el término "bloque de vídeo" para referirse a un bloque de árbol, es decir, una LCU o una CU, que incluye un nodo de codificación y unas PU y TU.

[0042] Una secuencia de vídeo incluye típicamente una serie de tramas o imágenes de vídeo. Un grupo de imágenes (GOP) comprende, en general, una serie de una o más de las imágenes de vídeo. Un GOP puede incluir datos sintácticos en una cabecera del GOP, una cabecera de una o más de las imágenes, o en otras ubicaciones, que describen una pluralidad de imágenes incluidas en el GOP. Cada fragmento de una imagen puede incluir datos sintácticos de fragmento que describen un modo de codificación para el fragmento respectivo. El codificador de vídeo actúa típicamente sobre bloques de vídeo dentro de fragmentos 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 de acuerdo con una norma de codificación especificada.

[0043] En un ejemplo, el HM admite la predicción en diversos tamaños de PU. Si se supone que el tamaño de una CU particular es 2Nx2N, el HM admite la predicción intra en tamaños de PU de 2Nx2N o NxN, y la predicción inter 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 predicción inter 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 está dividida, mientras que la otra dirección está dividida entre un 25 % y un 75 %. La parte de la CU correspondiente a la división del 25 % está indicada por una "n" seguida de una indicación de "arriba", "abajo", "izquierda" o "derecha". Así pues, por ejemplo, " $2N \times nU$ " se refiere a una CU de tamaño $2N \times 2N$ que está dividida horizontalmente, con una PU de tamaño $2N \times 0,5N$ encima y una PU de tamaño $2N \times 1,5N$ debajo.

[0044] En esta divulgación, " $N \times N$ " y " N por N " pueden usarse indistintamente para hacer referencia a las dimensiones de píxeles de un bloque de vídeo en cuanto a 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 una dirección vertical ($y = 16$) y 16 píxeles en una dirección horizontal ($x = 16$). Asimismo, un bloque de tamaño $N \times N$ tiene, en general, 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 se pueden disponer en filas y columnas. Además, no es necesario que los bloques tengan necesariamente el mismo número de píxeles en la dirección horizontal que en la dirección vertical. Por ejemplo, los bloques pueden comprender $N \times M$ píxeles, donde M no es necesariamente igual a N .

[0045] Tras la codificación intrapredictiva o interpredictiva, usando las PU de una CU, el codificador de vídeo 20 puede calcular datos residuales para las TU de la CU. Las PU pueden comprender datos de píxeles en el dominio espacial (también denominado dominio de píxel) y las TU pueden comprender coeficientes en el dominio de la transformada tras la aplicación de una transformada, por ejemplo, una transformada discreta de coseno (DCT), una transformada entera, una transformada de ondículas o una transformada conceptualmente similar, a los 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 formar las TU, incluidos los datos residuales para la CU y, a continuación, transformar las TU para generar coeficientes de transformada para la CU.

[0046] Tras cualquier transformada para generar coeficientes de transformada, el codificador de vídeo 20 puede realizar la cuantificación de los coeficientes de transformada. La cuantificación se refiere, en general, a un proceso en el que los coeficientes de transformada se cuantifican para reducir posiblemente la cantidad de datos usados para representar los coeficientes, proporcionando una compresión adicional. El proceso de cuantificación puede reducir la profundidad de bits asociada con algunos de, o todos, los coeficientes. Por ejemplo, un valor de n bits puede redondearse a la baja hasta un valor de m bits durante la cuantificación, donde n es mayor que m .

[0047] En algunos ejemplos, el codificador de vídeo 20 puede utilizar un orden de escaneado predefinido para escanear los coeficientes de transformada cuantificados, para producir un vector serializado que se pueda codificar por entropía. En otros ejemplos, el codificador de vídeo 20 puede realizar un escaneado adaptativo. Después de escanear los coeficientes de transformada cuantificados para formar un vector unidimensional, el codificador de vídeo 20 puede realizar la codificación por entropía del vector unidimensional, por ejemplo, de acuerdo con la codificación de longitud variable adaptativa al contexto (CAVLC), la codificación aritmética binaria adaptativa al contexto (CABAC), la codificación aritmética binaria adaptativa al contexto basada en la sintaxis (SBAC), la codificación por entropía de división en intervalos de probabilidad (PIPE) u otra metodología de codificación por entropía. El codificador de vídeo 20 también puede realizar la codificación por entropía de los elementos sintácticos asociados a los datos de vídeo codificados, para su uso por el descodificador de vídeo 30 en la descodificación de los datos de vídeo.

[0048] Para realizar la codificación CABAC, el codificador de vídeo 20 puede asignar un contexto, dentro de un modelo de contexto, a un símbolo a transmitir. El contexto se puede referir, por ejemplo, a si los valores contiguos del símbolo son distintos de cero o no. Para realizar la codificación CAVLC, el codificador de vídeo 20 puede seleccionar un código de longitud variable para un símbolo a transmitir. Las palabras de código en la codificación VLC se pueden construir de modo que los códigos relativamente más cortos correspondan a símbolos más probables, mientras que los códigos más largos correspondan a símbolos menos probables. De esta forma, el uso de la codificación VLC puede lograr un ahorro en bits con respecto, por ejemplo, al uso de palabras de código de igual longitud para cada símbolo que se va a transmitir. La determinación de la probabilidad se puede basar en un contexto asignado al símbolo.

[0049] De acuerdo con esta divulgación, el codificador de vídeo 20 del dispositivo de origen 12 puede escanear coeficientes de transformada de una unidad de datos de vídeo a nivel de hoja (por ejemplo, un nodo de hoja de un árbol cuádruple u otra estructura de datos) que incluye una matriz bidimensional de coeficientes de transformada (por ejemplo, que corresponden a píxeles de una imagen mostrada) en un vector unidimensional que representa los coeficientes de transformada. De acuerdo con las técnicas descritas en el presente documento, cuando se realiza tal escaneado, el codificador de vídeo 20 puede dividir la unidad a nivel de hoja de datos de vídeo en una pluralidad de subconjuntos de coeficientes de transformada. Para cada uno de los subconjuntos de la unidad a nivel de hoja, el codificador 20 puede determinar si se debe señalar o no a un descodificador como parte de un flujo de bits codificado por entropía, un elemento sintáctico que indica si el subconjunto incluye o no coeficientes distintos de cero. El codificador 20 puede determinar si se debe señalar o no el elemento sintáctico para un subconjunto particular basándose en determinar si la señalización del elemento sintáctico mejorará o no la eficiencia de codificación. Para determinar si la señalización del elemento sintáctico mejorará la eficiencia de la codificación, el codificador 20 puede aplicar una o más reglas, como se describe con más detalle a continuación. El codificador 20 puede generar un flujo de bits codificado por entropía que incluye el bloque de datos de vídeo. El flujo de bits codificado por entropía puede

ser leído y descodificado por un descodificador, para reconstruir la matriz bidimensional que representa la unidad de datos de vídeo a nivel de hoja.

5 **[0050]** La descodificación del coeficiente de transformada recíproca también puede realizarse mediante el descodificador de vídeo 30 del dispositivo de destino 14. Es decir, el descodificador de vídeo 30 puede asignar coeficientes de un vector unidimensional de coeficientes de transformada que representan un bloque de datos de vídeo a posiciones dentro de una matriz bidimensional de coeficientes de transformada, para reconstruir la matriz bidimensional de coeficientes de transformada. De acuerdo con las técnicas descritas en el presente documento, el descodificador 30 puede leer una matriz unidimensional que representa una unidad a nivel de hoja de datos de vídeo, y dividir la unidad a nivel de hoja en una pluralidad de subconjuntos de coeficientes de transformada. Para cada uno de los subconjuntos, el descodificador 30 puede determinar si descodifica o no los coeficientes de transformada del subconjunto. Por ejemplo, si el descodificador 30 no lee, en el flujo de bits codificado por entropía, un elemento sintáctico que indica si un subconjunto particular tiene o no coeficientes distintos de cero, entonces el descodificador 30 descodifica los coeficientes del subconjunto. Sin embargo, si el descodificador 30 lee dicho elemento sintáctico asociado con un subconjunto particular, el descodificador 30 puede determinar si descodifica o no los coeficientes de transformada del subconjunto basándose en el valor del elemento sintáctico. Por ejemplo, si el elemento sintáctico indica que el subconjunto incluye coeficientes distintos de cero, el descodificador 30 descodifica los coeficientes de transformada del subconjunto. Sin embargo, si el elemento sintáctico indica que el subconjunto no incluye ningún coeficiente distinto de cero, el descodificador 30 no descodifica los coeficientes de transformada del subconjunto.

20 **[0051]** Las técnicas descritas en el presente documento pueden mejorar la eficiencia de la codificación de vídeo. Por ejemplo, dividir un bloque de datos de vídeo en una pluralidad de subconjuntos de coeficientes de transformada y elementos sintácticos de señalización que indican si los subconjuntos incluyen o no coeficientes distintos de cero, como se describe en el presente documento, pueden reducir una cantidad de bits necesarios para representar datos de vídeo, lo cual puede mejorar el nivel de eficiencia del codificador/descodificador.

30 **[0052]** La FIG. 2 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo 20 que puede implementar las técnicas de predicción inter descritas en esta divulgación. El codificador de vídeo 20 puede realizar la intracodificación y la 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 el vídeo dentro de tramas o imágenes adyacentes de una secuencia de vídeo. La intramodalidad (modalidad I) puede referirse a cualquiera de varias modalidades de compresión de base espacial. Las intermodalidades, tales como la predicción unidireccional (modalidad P) o la bipredicción (modalidad B), pueden referirse a cualquiera de varias modalidades de compresión de base temporal.

35 **[0053]** En el ejemplo de la FIG. 2, el codificador 20 de vídeo incluye una unidad 35 de partición, un módulo 41 de predicción, una memoria 64 de imágenes de referencia, un sumador 50, un módulo 52 de transformada, una unidad 54 de cuantificación y una unidad 56 de codificación por entropía. El módulo de predicción 41 incluye una unidad de estimación de movimiento 42, una unidad de compensación de movimiento 44 y un módulo de predicción intra 46. Para la reconstrucción de bloques de vídeo, el codificador 20 de vídeo incluye además una unidad 58 de cuantificación inversa, un módulo 60 de transformada inversa y un sumador 62. También se puede incluir un filtro de eliminación de bloques (no mostrado en la FIG. 2) para filtrar fronteras de bloques, para eliminar distorsiones de efecto pixelado del vídeo reconstruido. Si se desea, el filtro de eliminación de bloques filtrará típicamente la salida del sumador 62. También se pueden usar filtros de bucle adicionales (en el bucle o tras el bucle), además del filtro de eliminación de bloques.

50 **[0054]** Como se muestra en la FIG. 2, el codificador de vídeo 20 recibe datos de vídeo, y la unidad de división 35 divide los datos en bloques de vídeo. Esta división también puede incluir la división en fragmentos, tejas u otras unidades mayores, así como la división de bloques de vídeo, por ejemplo, de acuerdo con una estructura de árbol cuádruple de las LCU y las CU. El codificador de vídeo 20 ilustra, en general, los componentes que codifican bloques de vídeo dentro de un fragmento de vídeo a codificar. El fragmento se puede dividir en múltiples bloques de vídeo (y, posiblemente, en conjuntos de bloques de vídeo denominados tejas). El módulo de predicción 41 puede seleccionar uno de entre una pluralidad de posibles modos de codificación, tal como uno de entre una pluralidad de modos de intracodificación, o uno de entre una pluralidad de modos de intercodificación, para el bloque de vídeo actual, basándose en resultados de errores (por ejemplo, la velocidad de codificación y el nivel de distorsión). El módulo de predicción 41 puede proporcionar el bloque intracodificado o intercodificado resultante al sumador 50 para generar datos de bloque residual, y al sumador 62 para reconstruir el bloque codificado para su uso como imagen de referencia.

60 **[0055]** El módulo de predicción intra 46, dentro del módulo de predicción 41, puede realizar la codificación intrapredictiva del bloque de vídeo actual con respecto a uno o más bloques contiguos de la misma trama o fragmento que el bloque actual que se vaya a codificar para proporcionar compresión espacial. La unidad 42 de estimación de movimiento y la unidad 44 de compensación de movimiento del módulo 41 de predicción realizan una codificación interpredictiva del bloque de vídeo actual en relación con uno o más bloques predictivos de una o más imágenes de referencia para proporcionar compresión temporal.

- 5 [0056] La unidad de estimación de movimiento 42 puede estar configurada para determinar la modalidad de predicción inter para un fragmento de vídeo, de acuerdo con un patrón predeterminado para una secuencia de vídeo. El patrón predeterminado puede designar fragmentos de vídeo en la secuencia como fragmentos P, fragmentos B o fragmentos GPB. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden estar sumamente integradas, pero se ilustran por separado con fines conceptuales. La estimación del movimiento, realizada por la unidad de estimación de movimiento 42, es el proceso de generación de vectores de movimiento, que estiman el movimiento para los 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 en relación con un bloque predictivo dentro de una imagen de referencia.
- 10 [0057] Un bloque predictivo es un bloque en el que se halla una estrecha correspondencia con la PU del bloque de vídeo a codificar, en términos de diferencia de píxeles, que puede determinarse mediante una suma de diferencias absolutas (SAD), una suma de diferencias al cuadrado (SSD) u otras métricas de diferencia. En algunos ejemplos, el codificador de vídeo 20 puede calcular valores para posiciones fraccionarias de píxeles 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 fraccionarias de píxel 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 completo y las posiciones de píxel fraccionario, y enviar un vector de movimiento con una precisión de píxel fraccionario.
- 20 [0058] 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 a partir de una primera lista (lista 0) de imágenes de referencia o una segunda lista (lista 1) de imágenes de referencia, cada una de las cuales identifica una o más imágenes de referencia almacenadas en una memoria de imágenes de referencia 64. La unidad de estimación de movimiento 42 envía el vector de movimiento calculado a la unidad de codificación por entropía 56 y a la unidad de compensación de movimiento 44.
- 25 [0059] La compensación de movimiento, realizada por la unidad de compensación de movimiento 44, puede implicar extraer o generar el bloque predictivo basándose en el vector de movimiento determinado mediante estimación de movimiento, realizando posiblemente interpolaciones hasta una precisión de subpíxel. Tras recibir el vector de movimiento para la PU del bloque de vídeo actual, la unidad de compensación de movimiento 44 puede localizar el bloque predictivo al que apunta el vector de movimiento en una de las listas de imágenes de referencia. El codificador de vídeo 20 forma un bloque de vídeo residual restando los valores de píxeles del bloque predictivo a los valores de píxeles del bloque de vídeo actual que se está codificando, formando valores de diferencias de píxel. Los valores de diferencias de píxel forman datos residuales para el bloque, y pueden incluir componentes de diferencia, tanto de luma como de croma. El sumador 50 representa el componente o los componentes que realizan esta operación de resta. La unidad de compensación de movimiento 44 también puede generar elementos sintácticos asociados con los bloques de vídeo y con el fragmento de vídeo para su uso por el descodificador de vídeo 30 en la descodificación de los bloques de vídeo del fragmento de vídeo.
- 30 [0060] Después de que la unidad de compensación de movimiento 44 genere el bloque predictivo para el bloque de vídeo actual, el codificador de vídeo 20 forma un bloque de vídeo residual al restar el bloque predictivo del bloque de vídeo actual. Los datos de vídeo residuales del bloque residual pueden incluirse en una o más TU y aplicarse al módulo de transformada 52. El módulo de transformada 52 transforma los datos de vídeo residuales en coeficientes de transformada residuales mediante una transformada, tal como una transformada de coseno discreta (DCT) o una transformada conceptualmente similar. El módulo de transformada 52 puede convertir los datos de vídeo residuales de un dominio de píxel en un dominio de transformada, tal como un dominio de frecuencia.
- 35 [0061] El módulo 52 de transformada puede enviar los coeficientes de transformada resultantes a la unidad 54 de cuantificación. La unidad de cuantificación 54 cuantifica los coeficientes de transformada para reducir más la velocidad de bits. El proceso de cuantificación puede reducir la profundidad de bits asociada con algunos de, o todos, los coeficientes. El grado de cuantificación se puede modificar ajustando un parámetro de cuantificación.
- 40 [0062] Tras la cuantificación, la unidad de codificación por entropía 56 realiza la codificación por entropía de los coeficientes de transformada cuantificados. Por ejemplo, la unidad de codificación por entropía 56 puede realizar una codificación de longitud variable adaptativa al contexto (CAVLC), una codificación aritmética binaria adaptativa al contexto (CABAC), una codificación aritmética binaria adaptativa al contexto basada en la sintaxis (SBAC), una codificación por entropía por división de intervalos de probabilidad (PIPE) u otra metodología o técnica de codificación por entropía. Tras la codificación por entropía por la unidad de codificación por entropía 56, el flujo de bits codificado se puede transmitir al descodificador de vídeo 30, o archivarse para su posterior transmisión o recuperación por el descodificador de vídeo 30. La unidad de codificación por entropía 56 también puede realizar la codificación por entropía de los vectores de movimiento y los otros elementos sintácticos para el fragmento de vídeo actual que se está codificando. En algunos ejemplos, la unidad de codificación por entropía 56 puede realizar un escaneado de la matriz que incluye los coeficientes de transformada cuantificados para generar un vector unidimensional de coeficientes de transformada de un flujo de bits codificado por entropía.
- 45
- 50
- 55
- 60
- 65

[0063] De acuerdo con las técnicas de esta divulgación, cuando se escanea una matriz de coeficientes de transformada para generar un vector unidimensional, la unidad de codificación por entropía 56 puede dividir una unidad de datos de vídeo a nivel de hoja, como un nodo hijo no dividido de una estructura de árbol cuádruple como se describió anteriormente, en una pluralidad de subconjuntos de coeficientes de transformada que son más pequeños que la unidad a nivel de hoja. Por ejemplo, la unidad de codificación por entropía 56 puede dividir la unidad a nivel de hoja de datos de vídeo en una pluralidad de subconjuntos, cada uno de los cuales comprende una forma geométrica dentro de la unidad a nivel de hoja. En algunos ejemplos, la unidad de codificación por entropía 56 puede dividir los coeficientes de transformada de la unidad a nivel de hoja en disposiciones de datos de vídeo de forma rectangular más pequeñas. En otros ejemplos, la unidad de codificación por entropía 56 puede dividir los coeficientes de transformada de la unidad a nivel de hoja en subconjuntos triangulares de coeficientes de transformada. En otros ejemplos más, la unidad de codificación por entropía 56 puede dividir los coeficientes de transformada de la unidad a nivel de hoja en subconjuntos que tienen otras formas, o incluso subconjuntos que no corresponden a ninguna forma geométrica particular. En cambio, la unidad de codificación por entropía 56 puede dividir la pluralidad de coeficientes de transformada de la unidad a nivel de hoja de acuerdo con un orden de escaneado de los coeficientes de transformada, que pueden ser fijos o adaptativos. Por ejemplo, la unidad de codificación por entropía 56 puede dividir la pluralidad de coeficientes de transformada de las unidades a nivel de hoja en una pluralidad de subconjuntos de coeficientes de transformada, basándose en un orden de escaneado (por ejemplo, un orden de escaneado adaptativo o fijo) de los coeficientes de transformada. Los subconjuntos pueden tener números de coeficientes similares o diferentes en diferentes ejemplos.

[0064] De acuerdo con las técnicas descritas en el presente documento, una vez que la unidad de coeficientes de transformada a nivel de hoja se ha dividido en una pluralidad de subconjuntos, la unidad de codificación por entropía 56 puede generar, para al menos parte de la pluralidad de subconjuntos, un elemento sintáctico que indica si el subconjunto respectivo incluye algún coeficiente distinto de cero. En algunos ejemplos, la unidad de codificación por entropía 56 puede generar dicho elemento sintáctico para cada subconjunto de la pluralidad de subconjuntos de coeficientes de transformada. En otros ejemplos, la unidad de codificación por entropía 56 puede determinar selectivamente si generar o no dicho elemento sintáctico, para cada subconjunto de la pluralidad de subconjuntos. La decisión de generar o no dicho elemento sintáctico puede basarse en si dicha señalización proporciona una compresión adicional o una eficiencia de codificación mejorada. Si es así, el elemento sintáctico puede generarse, pero si no, tanto el codificador 20 como el descodificador 30 pueden programarse para saber que la indicación no está señalizada en ese caso.

[0065] De acuerdo con las técnicas de esta divulgación, para determinar si generar o no dicho elemento sintáctico mejorará la compresión o la eficiencia de codificación, la unidad de codificación por entropía 56 puede aplicar una o más reglas, como se describe con más detalle a continuación. Como un ejemplo, la unidad de codificación por entropía 56 puede determinar si generar el elemento sintáctico para un subconjunto particular basado en un número de coeficientes potenciales distintos del cero del subconjunto. Por ejemplo, la unidad de codificación por entropía 56 puede determinar si generar el elemento sintáctico basándose en la comparación de un número de coeficientes potenciales distintos de cero del subconjunto con un umbral.

[0066] Como un ejemplo de dicha técnica, para determinar el número de coeficientes potenciales distintos de cero de un subconjunto, la unidad de codificación por entropía 56 determina un número de coeficientes del subconjunto que tienen una posición anterior a un último coeficiente distinto de cero de la unidad a nivel de hoja y compara el número determinado de coeficientes con un umbral $thNoCoeff$, como se muestra en el ejemplo de la FIG. 6 y se describe con más detalle a continuación. De acuerdo con este ejemplo, si el número determinado de coeficientes potenciales distintos de cero es mayor que el umbral $thNoCoeff$, la unidad de codificación por entropía 56 genera un elemento sintáctico que indica si el subconjunto incluye o no coeficientes distintos de cero. Sin embargo, si el número determinado de coeficientes potenciales distintos de cero es menor o igual que el umbral $thNoCoeff$, la unidad de codificación por entropía 56 puede no generar el elemento sintáctico. Como ejemplo, la unidad de codificación por entropía 56 determina si se debe señalar un elemento sintáctico que indica si un subconjunto particular incluye algún coeficiente distinto de cero basado en el pseudocódigo del Ejemplo 1 siguiente:

Ejemplo 1 $noCoeff[xS] [yS] > thNoCoeff$

[0067] donde $noCoeff[xS] [yS]$ indica una cantidad de coeficientes potenciales distintos de cero de un subconjunto, y $thNoCoeff$ es un valor umbral.

[0068] En otros ejemplos, la unidad de codificación por entropía 56 puede determinar si se debe indicar o no el elemento sintáctico que indica si un subconjunto de coeficientes de transformada incluye cualquier coeficiente distinto de cero basado en otras reglas. Por ejemplo, la unidad de codificación por entropía 56 puede determinar si se debe señalar o no el elemento sintáctico para un subconjunto basado en un número medio de coeficientes distintos de cero del subconjunto, como se describe con más detalle a continuación con respecto al ejemplo de la FIG. 7. De acuerdo con este ejemplo, a medida que se codifican unidades de datos de vídeo a nivel de hoja, la unidad de codificación por entropía 56 puede recopilar y almacenar estadísticas que indican con qué frecuencia los coeficientes en las posiciones dentro de cada subconjunto no son cero. Basándose en tales estadísticas almacenadas, la unidad

de codificación por entropía 56 puede determinar si generar o no el elemento sintáctico. Por ejemplo, la unidad de codificación por entropía 56 puede comparar un número medio de coeficientes distintos de cero para un subconjunto con un umbral $thAvergCoeff$. Si el número medio de coeficientes distintos de cero del subconjunto es menor que el umbral, la unidad de codificación por entropía 56 genera el elemento sintáctico. Sin embargo, si el número medio de coeficientes distintos de cero del subconjunto es mayor o igual al umbral $thAvergCoeff$, la unidad de codificación por entropía 56 no genera el elemento sintáctico basándose en el supuesto de que en este caso es muy probable que el subconjunto contenga al menos un coeficiente distinto de cero.

[0069] Como otro ejemplo, para un subconjunto que contiene el último coeficiente distinto de cero (en relación con un escaneado en zig-zag, un primer coeficiente distinto de cero de un escaneado en zig-zag inverso) de la unidad a nivel de hoja, el hecho de que el subconjunto contiene el último coeficiente distinto de cero significa que ese subconjunto debe incluir al menos un coeficiente distinto de cero y, por lo tanto, se puede suponer que cualquier subconjunto que incluya el último coeficiente distinto de cero debe incluir al menos un coeficiente distinto de cero. En consecuencia, no es necesario generar el elemento sintáctico para un subconjunto que incluye el último coeficiente distinto de cero, ya que se puede suponer que este subconjunto incluye al menos un coeficiente distinto de cero.

[0070] Como ejemplo, la unidad de codificación por entropía 56 puede determinar si se debe señalar un elemento sintáctico que indica si un subconjunto particular incluye algún coeficiente distinto de cero basado en el pseudocódigo del ejemplo 2 siguiente:

Ejemplo 2 $(noCodedCoeff[xS][yS] + noSubBlks[xS][yS]/2) / noSubBlks < thAvergCoeff$

[0071] Aquí $noCodedCoeff[xS][yS]$ indica el número de coeficientes distintos de cero en un subconjunto.

[0072] De acuerdo con el pseudocódigo del Ejemplo 1, la unidad de codificación por entropía 56 inicializa los valores en la matriz variable $noCoeff[xS][yS]$ a 0 antes de codificar una unidad a nivel de hoja del bloque de datos de vídeo y asignar valores a la matriz basándose en un valor de un elemento sintáctico $lastPos$, que indica una última posición de coeficiente diferente de cero dentro de una unidad de datos de vídeo a nivel de hoja:

```
for (pos=0; pos<= lastPos; pos++) {
    xC= ScanOrder [0][pos];
    yC= ScanOrder [1][pos];
    noCoeff[xC/M][yC/M]++; }
```

[0073] De acuerdo con el pseudocódigo del Ejemplo 2, después de que se codifica la unidad a nivel de hoja, la unidad de codificación por entropía 56 puede actualizar las matrices variables $noSubBlks$ y $noCodedCoeff$ de la siguiente manera:

```
for (pos=0; pos<= lastPos; pos++){
    xC= ScanOrder [0][pos];
    yC= ScanOrder [1][pos];
    if (transCoeffLevel[xC][yC] != 0) {
        noCodedCoeff[xC/M][yC/M]++;
    }
}
for (xS=0; xS<N/M; xS++){
    for (yS=0; yS<N/M; yS++){
        if (noCoeff[xS][yS]>0){
            noSubBlks[xS][yS]++; } }
```

[0074] Al utilizar el pseudocódigo del Ejemplo 2 anterior, la unidad de codificación por entropía 56 puede determinar si se debe indicar o no un elemento sintáctico que indique si un subconjunto de coeficientes de transformada de una unidad a nivel de hoja incluye cualquier coeficiente distinto de cero, basado en un número medio de coeficientes distintos de cero del subconjunto para unidades de datos de vídeo a nivel de hoja previamente codificadas.

[0075] La unidad 58 de cuantificación inversa y el módulo 60 de transformada inversa aplican una cuantificación inversa y una transformación inversa, respectivamente, para reconstruir el bloque residual en el dominio del píxel, para su uso posterior como bloque de referencia de una imagen 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 imágenes de referencia dentro de una de las listas de imágenes de referencia. La unidad de compensación de movimiento 44 también puede aplicar uno o más filtros de interpolación al bloque residual reconstruido para calcular valores fraccionarios de píxeles, para su uso en la estimación de movimiento. El sumador 62 añade el bloque residual reconstruido al bloque de predicción compensado por movimiento, generado por la unidad de compensación de movimiento 44 para generar un bloque de referencia para su almacenamiento en la memoria de imágenes de referencia 64. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden usar

el bloque de referencia como bloque de referencia para realizar la predicción inter de un bloque en una trama o imagen de vídeo posterior.

[0076] La FIG. 3 es un diagrama de bloques que ilustra un ejemplo de descodificador de vídeo 30 que puede implementar las técnicas de predicción inter descritas en esta divulgación. En el ejemplo de la FIG. 3, el descodificador de vídeo 30 incluye una unidad de descodificación por entropía 80, un módulo de predicción 81, una unidad de cuantificación inversa 86, una unidad de transformación inversa 88, un sumador 90, y una memoria de imágenes de referencia 92. El módulo de predicción 81 incluye la unidad de compensación de movimiento 82 y el módulo de predicción intra 84. En algunos ejemplos, el descodificador de vídeo 30 puede realizar un pase de descodificación que, en general, es recíproco al pase de codificación descrito con respecto al codificador de vídeo 20 de la FIG. 2.

[0077] Durante el proceso de descodificación, el descodificador 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, desde el codificador de vídeo 20. La unidad de descodificación por entropía 80 del descodificador de vídeo 30 descodifica por entropía el flujo de bits para generar coeficientes cuantificados, vectores de movimiento y otros elementos sintácticos. La unidad 80 de descodificación por entropía envía los vectores de movimiento y otros elementos sintácticos al módulo 81 de predicción. El descodificador de vídeo 30 puede recibir los elementos sintácticos en el nivel de fragmento de vídeo y/o el nivel de bloque de vídeo.

[0078] La unidad de descodificación por entropía 80 puede leer un vector unidimensional de coeficientes de transformada descodificados por la unidad de descodificación por entropía, y reconstruir una matriz bidimensional de coeficientes de transformada a partir del vector unidimensional. De acuerdo con las técnicas descritas en el presente documento, la unidad de descodificación por entropía 80 puede leer una matriz unidimensional que representa una unidad de datos de vídeo a nivel de hoja y dividir la unidad a nivel de hoja en una pluralidad de subconjuntos de coeficientes de transformada. Los subconjuntos de coeficientes de transformada pueden tener una forma o disposición rectangular, triangular o de cualquier otro tipo. Para cada uno de los subconjuntos, la unidad de descodificación por entropía inversa 80 determina si descodificar o no los coeficientes de transformada del subconjunto. Por ejemplo, si la unidad de descodificación por entropía 80 no lee un elemento sintáctico que indica si un subconjunto particular tiene o no coeficientes distintos de cero, entonces la unidad de cuantificación inversa 86 descodifica los coeficientes del subconjunto. Sin embargo, si la unidad de descodificación por entropía 80 lee dicho elemento sintáctico asociado con un subconjunto particular, la unidad de descodificación por entropía 80 puede determinar si descodifica o no los coeficientes de transformada del subconjunto basándose en el valor del elemento sintáctico. Por ejemplo, si el elemento sintáctico indica que el subconjunto incluye coeficientes distintos de cero, la unidad de descodificación por entropía 80 descodifica los coeficientes de transformada del subconjunto. Sin embargo, si el elemento sintáctico indica que el subconjunto no incluye ningún coeficiente distinto de cero, la unidad de descodificación por entropía 80 no descodifica los coeficientes de transformada del subconjunto.

[0079] Cuando el fragmento de vídeo se codifica como un fragmento intracodificado (I), el módulo 84 de predicción intra del módulo 81 de predicción puede generar datos de predicción para un bloque de vídeo del fragmento de vídeo actual, basándose en un modo de predicción intra señalado y datos de bloques descodificados previamente 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 82 de compensación de movimiento del módulo 81 de predicción genera bloques predictivos para un bloque de vídeo del fragmento de vídeo actual, basándose en los vectores de movimiento y otros elementos sintácticos recibidos desde la unidad 80 de descodificación por entropía. Los bloques predictivos se pueden generar a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El descodificador de vídeo 30 puede construir las listas de tramas de referencia, Lista 0 y Lista 1, utilizando técnicas de construcción predeterminadas, basándose en imágenes de referencia almacenadas en la memoria de imágenes de referencia 92.

[0080] La unidad de compensación de movimiento 82 determina la información de predicción para un bloque de vídeo del fragmento de vídeo actual, analizando sintácticamente los vectores de movimiento y otros elementos sintácticos, y usa la información de predicción para generar los bloques predictivos para el bloque de vídeo actual que se está descodificando. Por ejemplo, la unidad de compensación de movimiento 82 usa algunos de los elementos sintácticos recibidos para determinar una modalidad de predicción (por ejemplo, predicción intra o predicción inter) usada para codificar los bloques de vídeo del fragmento de vídeo, un tipo de fragmento de predicción inter (por ejemplo, un fragmento B, un fragmento P o un fragmento GPB), 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, el estado de predicción inter para cada bloque de vídeo intercodificado del fragmento y otra información para descodificar los bloques de vídeo en el fragmento de vídeo actual.

[0081] La unidad de compensación de movimiento 82 también puede realizar la interpolación basándose en filtros de interpolación. La unidad de compensación de movimiento 82 puede usar filtros de interpolación como los usados por el codificador de vídeo 20 durante la codificación de los bloques de vídeo, para calcular valores interpolados para píxeles fraccionarios de los bloques de referencia. En este caso, la unidad de compensación de movimiento 82 puede determinar los filtros de interpolación usados por el codificador de vídeo 20 a partir de los elementos sintácticos recibidos y usar los filtros de interpolación para generar bloques predictivos. La unidad de cuantificación inversa 86

cuantifica inversamente, es decir, decuantifica, los coeficientes de transformada cuantificados proporcionados en el flujo de bits y descodificados por la unidad de descodificación por entropía 80.

5 **[0082]** En algunos ejemplos, el proceso de cuantificación inversa puede incluir el uso de un parámetro de cuantificación calculado por el codificador de vídeo 20 para cada bloque de vídeo en el fragmento de vídeo, con el fin de determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa que se debería aplicar. El módulo 88 de transformada inversa aplica una transformada inversa, por ejemplo, una DCT inversa, una transformada entera inversa o un proceso de transformada inversa conceptualmente similar, a los coeficientes de transformada, con el fin de generar bloques residuales en el dominio del píxel.

10 **[0083]** Después de que la unidad 82 de compensación de movimiento genere el bloque predictivo para el bloque de vídeo actual basándose en los vectores de movimiento y otros elementos sintácticos, el descodificador 30 de vídeo forma un bloque de vídeo descodificado sumando los bloques residuales procedentes del módulo 88 de transformada inversa a los correspondientes bloques predictivos generados por la unidad 82 de compensación de movimiento. El sumador 90 representa el componente o los componentes que realizan esta operación de suma. Si se desea, también se puede aplicar un filtro de eliminación de bloques para filtrar los bloques descodificados, a fin de eliminar distorsiones de efecto pixelado. También se pueden usar otros filtros de bucle (bien en el bucle de codificación o bien después del bucle de codificación) para allanar las transiciones de píxeles o mejorar de otro modo la calidad del vídeo. Los bloques de vídeo descodificados en una trama o imagen dada se almacenan a continuación en la memoria de imágenes de referencia 92, que almacena imágenes de referencia usadas para una compensación de movimiento posterior. La memoria de imágenes de referencia 92 almacena también vídeo descodificado para su presentación posterior en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la FIG. 1.

25 **[0084]** La FIG. 4 es un diagrama conceptual que representa un ejemplo de una unidad 410 a nivel de hoja dividida en una pluralidad de subconjuntos de coeficientes de transformada consistentes con las técnicas descritas en el presente documento. La unidad a nivel de hoja 410 representada en la FIG. 4 puede comprender un nodo hijo no dividido de datos de vídeo, por ejemplo, un nodo hoja de una estructura de árbol cuádruple (RQT). Como se muestra en la FIG. 4, cada una de la pluralidad de subconjuntos 420 incluye una pluralidad de coeficientes de transformada 412 (dieciséis coeficientes de transformada en el ejemplo de la FIG. 1), que pueden ser o no coeficientes distintos de cero con una amplitud mayor que cero. Como también se muestra en la FIG. 4, un último coeficiente distinto de cero (relativo a un escaneado en zig-zag, un primer coeficiente distinto de cero de un escaneado en zig-zag inverso) de la unidad a nivel de hoja 410 tiene una posición (7, 7) dentro del subconjunto (1,1) de la unidad a nivel de hoja 410.

35 **[0085]** Como se describió anteriormente, una vez que el codificador de vídeo 20 ha dividido la unidad a nivel de hoja 410 en una pluralidad de subconjuntos de coeficientes de transformada 420, el codificador de vídeo 20 puede determinar si generar un elemento sintáctico que indique si el subconjunto incluye o no coeficientes distintos de cero. Por ejemplo, el codificador de vídeo 20 puede determinar si generar o no el elemento sintáctico asociado con un bloque particular basándose en una serie de coeficientes potenciales distintos de cero en el bloque.

40 **[0086]** De acuerdo con el ejemplo de la FIG. 4, un coeficiente en la posición (7, 7) es un último coeficiente distinto de cero de la unidad a nivel de hoja 420, lo cual significa que todos los coeficientes de transformada después de la posición (7, 7) de acuerdo con un orden de escaneado en zig-zag (los coeficientes sombreados en el ejemplo de la FIG. 4) tienen una amplitud de cero. De acuerdo con un ejemplo, el codificador de vídeo 20 puede determinar varios coeficientes potenciales distintos de cero en un subconjunto basado en una última posición de coeficiente diferente de cero dentro de la unidad a nivel de hoja 410.

50 **[0087]** El codificador de vídeo 20 puede, para cada subconjunto respectivo de la unidad 410 a nivel de hoja, determinar el número potencial de coeficientes distintos de cero (por ejemplo, coeficientes de escaneado en zig-zag inverso que siguen al último coeficiente distinto de cero del escaneado en ubicación (7, 7) del subconjunto). Por ejemplo, el codificador de vídeo 20 puede determinar que los dieciséis coeficientes de cada uno de los subconjuntos (0, 0), (0, 1), (1, 0), (1, 1) y (2, 0) representados en la FIG. 4 son coeficientes potencialmente distintos de cero, porque todos los coeficientes de los subconjuntos están antes del último coeficiente distinto de cero en el escaneado. El codificador de vídeo también puede determinar que quince coeficientes del subconjunto (0, 2) pueden ser potencialmente coeficientes distintos de cero, seis coeficientes de cada uno de los subconjuntos (2, 1) y (3, 0) pueden ser distintos de cero coeficientes y tres coeficientes de los subconjuntos (1, 1) y (0, 3) pueden ser potencialmente coeficientes distintos de cero.

60 **[0088]** El codificador 20 puede determinar si generar un elemento sintáctico para los subconjuntos 420 representados en la FIG. 4 basado en el número determinado de coeficientes potenciales distintos de cero para cada subconjunto. Por ejemplo, el codificador 20 puede comparar el número determinado de coeficientes potenciales distintos de cero para cada subconjunto con un umbral $thNoCoeff$. Si el número determinado de coeficientes potenciales distintos de cero es mayor que el umbral, el codificador 20 puede generar el elemento sintáctico que indica si el subconjunto incluye o no coeficientes distintos de cero. Sin embargo, si el número determinado de coeficientes potenciales distintos de cero es menor o igual que el umbral, el codificador 20 puede no generar el elemento sintáctico.

65

[0089] De acuerdo con un ejemplo específico, el umbral $thNoCoeff$ puede tener un valor de cinco (5). Con referencia al ejemplo de la FIG. 4, el codificador 20 señalaría el elemento sintáctico que indica si los subconjuntos (0, 0), (0, 1), (1, 0), (1, 1), (0, 2), (2, 0), (2, 1) y (3, 0) incluyen coeficientes distintos de cero, porque cada uno de estos subconjuntos 420 incluye más coeficientes potenciales distintos de cero que el valor umbral $thNoCoeff$ de cinco. Sin embargo, para los subconjuntos (1,2), (0,3), (3,1), (2,2), (1,3), (3,2), (2,3) y (3,3), el codificador de vídeo 20 no generaría el elemento sintáctico, porque los subconjuntos incluyen menos de cinco coeficientes potenciales distintos de cero. Por ejemplo, el codificador de vídeo 20 puede codificar los coeficientes de transformada de los subconjuntos (1,1) y (0, 2) y determinar si los subconjuntos incluyen algún coeficiente distinto de cero. Si el codificador de vídeo 20 determina que cualquiera de los subconjuntos (1,2) y (0, 3) incluyen coeficientes distintos de cero, el codificador de vídeo puede generar un elemento sintáctico con un valor de uno (1) asociado con el subconjunto. De lo contrario, si el codificador de vídeo 20 determina que cualquiera de los subconjuntos (1,1) y (0, 2) no incluyen ningún valor distinto de cero, el codificador de vídeo 20 también puede generar un elemento sintáctico con un valor de cero (0) asociado con el subconjunto. El codificador de vídeo 20 también puede generar un elemento sintáctico con un valor de (0) o (1) asociado con cada uno de los subconjuntos (0, 0), (0, 1), (1, 0), (2, 0), (2, 1) y (3, 0).

[0090] Los elementos sintácticos que indican si los subconjuntos (0, 0), (0, 1), (1, 0), (1, 1), (0, 2), (2, 0), (2, 1), y (3, 0) tienen coeficientes distintos de cero (por ejemplo, los elementos sintácticos separados definidos para cada subconjunto respectivo) pueden ser enviados por el codificador de vídeo 20 como parte de un flujo de bits codificado por entropía, y leídos por el descodificador 30. Como parte de la reconstrucción de la unidad a nivel de hoja 410, el descodificador 30 puede dividir los datos que representan la unidad 410 en una pluralidad de subbloques. Para cada subbloque, el descodificador 30 puede determinar si el flujo de bits codificado por entropía incluye un elemento sintáctico que indica si el subconjunto incluye coeficientes distintos de cero. Si el descodificador 30 lee un subbloque que no incluye dicho elemento sintáctico, el descodificador 30 descodifica los coeficientes del subconjunto. Sin embargo, si el descodificador 30 lee dicho elemento sintáctico asociado con un subconjunto, el descodificador 30 puede usar el valor del elemento sintáctico para determinar si descodifica los coeficientes del subconjunto. Por ejemplo, de acuerdo con el ejemplo expuesto anteriormente, si el descodificador 30 lee un valor de elemento sintáctico de uno (1) para el subconjunto (1,1), el descodificador puede descodificar el subconjunto. Sin embargo, si el descodificador 30 lee un valor de elemento sintáctico de cero (0) para el subconjunto (1,1), el descodificador 30 no descodifica los coeficientes de transformada del subbloque.

[0091] La FIG. 5 es un diagrama de flujo que ilustra un ejemplo de un procedimiento de datos de vídeo de codificación consistente con uno o más aspectos de esta divulgación. El procedimiento de la FIG. 5 se describe como realizado por el codificador 20 representado en las FIGS. 1 y 2; sin embargo, se puede usar cualquier otro dispositivo para realizar el procedimiento de la FIG. 5.

[0092] Como se muestra en la FIG. 5, el codificador 20 divide un bloque de hoja de datos de vídeo en una pluralidad de subconjuntos, cada uno de los cuales incluye un coeficiente de transformada múltiple (501). Como un ejemplo, la pluralidad de subconjuntos puede comprender subconjuntos de forma rectangular como se representa en el ejemplo de la FIG. 4, o cualquier otra forma geométrica o patrón. La unidad de datos de vídeo a nivel de hoja puede comprender una unidad de codificación a nivel más bajo de un estándar de codificación de vídeo, tal como un nodo hijo no dividido de una estructura de árbol cuádruple como se describió anteriormente.

[0093] Como también se muestra en la FIG. 5, el codificador 20 genera, para al menos uno de la pluralidad de subconjuntos, un elemento sintáctico que indica si el subconjunto incluye o no coeficientes distintos de cero (502). En algunos ejemplos, el codificador 20 puede generar dicho elemento sintáctico asociado con cada una de la pluralidad de subconjuntos. En otros ejemplos, el codificador 20 puede determinar selectivamente si generar o no dicho elemento sintáctico para cada uno de los subconjuntos. Por ejemplo, el codificador 20 puede determinar selectivamente si generar el elemento sintáctico basándose en si la generación del elemento sintáctico mejorará la eficiencia de codificación y/o la compresión. La sobrecarga asociada con la indicación de que un subconjunto particular en realidad incluye coeficientes distintos de cero puede degradar la compresión en algunos casos, y en estos casos, dicha señalización de sobrecarga puede evitarse, y el codificador y descodificador pueden codificar/descodificar el subconjunto sin cualquier determinación de si el subconjunto incluye o no coeficientes distintos de cero.

[0094] En algunos ejemplos, el codificador 20 puede determinar si generar el elemento sintáctico mejorará la eficiencia de codificación basándose en una serie de coeficientes potenciales distintos de cero en un subconjunto como se describe con más detalle a continuación con respecto a la FIG. 6, o basándose en un número medio de coeficientes de cada subconjunto como se describe con más detalle a continuación, como se describe con más detalle a continuación con respecto a la FIG. 7. Si el codificador 20 genera el elemento sintáctico asociado con un subconjunto, el codificador 20 puede generar el elemento sintáctico como parte de un flujo de bits codificado por entropía. El flujo de bits codificado por entropía puede ser leído por un descodificador 30 como se representa en la FIG. 3, y el descodificador 30 puede usar el elemento sintáctico para descodificar la unidad a nivel de hoja, como se describe con más detalle con respecto a la FIG. 8 a continuación.

[0095] La FIG. 6 es un diagrama de flujo que ilustra un ejemplo de un procedimiento de datos de vídeo de codificación consistente con uno o más aspectos de esta divulgación. El procedimiento de la FIG. 6 se describe como realizado por el codificador 20 representado en la FIG. 2; sin embargo, se puede usar cualquier dispositivo para realizar las

técnicas de la FIG. 6. De acuerdo con el ejemplo de la FIG. 6, el codificador 20 determina si generar un elemento sintáctico asociado con un subconjunto de coeficientes de transformada mejorará la eficiencia de codificación basándose en varios coeficientes potenciales distintos de cero para el subconjunto.

5 **[0096]** Como se muestra en el ejemplo de la FIG. 6, el codificador 20 divide una unidad de datos de vídeo a nivel de hoja en una pluralidad de subconjuntos de coeficientes de transformada (601). Como un ejemplo, la pluralidad de subconjuntos puede comprender subconjuntos de forma rectangular como se representa en el ejemplo de la FIG. 4, o cualquier otra forma geométrica o patrón. La unidad de datos de vídeo a nivel de hoja puede comprender una unidad de codificación a nivel más bajo de un estándar de codificación de vídeo, tal como un nodo hijo no dividido de una estructura de árbol cuádruple como se describió anteriormente.

15 **[0097]** De acuerdo con el ejemplo, de la FIG. 6, el codificador de vídeo 20 determina si generar o no un elemento sintáctico asociado con cada una de la pluralidad de subconjuntos basándose en una serie de coeficientes potenciales distintos de cero de cada subconjunto. Por ejemplo, como se muestra en la FIG. 6, el codificador 20 determina, para cada subconjunto, un número de coeficientes potenciales distintos de cero del subconjunto (602). Para determinar el número de coeficientes potenciales distintos de cero, el codificador 20 puede determinar cuántos coeficientes de cada subconjunto tienen una posición antes del último coeficiente distinto de cero de la unidad a nivel de hoja, como se describió anteriormente con respecto al ejemplo de la FIG. 4.

20 **[0098]** Como también se muestra en la FIG. 6, el codificador 20 puede comparar el número determinado de coeficientes potenciales distintos de cero con un valor umbral $thNoCoeff$ (603). Como también se muestra en la FIG. 6, si el número determinado de coeficientes potenciales distintos de cero es mayor que el valor umbral $thNoCoeff$, el codificador 20 genera el elemento sintáctico asociado con el subconjunto (604). Sin embargo, si el número determinado de coeficientes potenciales distintos de cero es menor o igual que el valor umbral $thNoCoeff$, el codificador 20 no genera el elemento sintáctico asociado con el subconjunto (605). De acuerdo con estos ejemplos, si el número de coeficientes potenciales distintos de cero es menor o igual que el valor umbral $thNoCoeff$, se puede suponer que se debe realizar la descodificación/codificación/descodificación del subconjunto. De esta manera, el elemento sintáctico no se genera para aquellos subconjuntos que incluyen muy pocos coeficientes potenciales distintos de cero, lo cual puede reducir una cantidad de bits de información señalados por el codificador 20 para representar la unidad de datos de vídeo a nivel de hoja. Esto se debe a que cuando un subconjunto contiene muy pocos coeficientes potenciales distintos de cero, el número de bits necesarios para señalar esos coeficientes distintos de cero es relativamente pequeño, lo cual a su vez resulta en una sobrecarga relativamente mayor al indicar el elemento sintáctico que indica si el subconjunto tiene coeficientes distintos de cero. Como resultado, es mejor no enviar señales al elemento sintáctico y, en lugar de ello, codificar los coeficientes del subconjunto directamente cuando el subconjunto contiene muy pocos coeficientes potenciales distintos de cero (es decir, menos que $thNoCoeff$).

35 **[0099]** La FIG. 7 es un diagrama de flujo que ilustra otro ejemplo de un procedimiento de codificación de datos de vídeo consistente con uno o más aspectos de esta divulgación. El procedimiento de la FIG. 7 se describe como realizado por el codificador 20 representado en la FIG. 2; sin embargo, se puede usar cualquier dispositivo para realizar las técnicas de la FIG. 7. De acuerdo con el ejemplo de la FIG. 7, el codificador 20 determina si generar un elemento sintáctico mejorará la eficiencia de codificación basándose en un número medio de coeficientes distintos de cero para cada subconjunto.

45 **[0100]** Como se muestra en la FIG. 7, el codificador 20 divide una unidad de datos de vídeo a nivel de hoja en una pluralidad de subconjuntos de coeficientes de transformada (701). Como también se muestra en la FIG. 7, para cada subconjunto, el codificador 20 determina un número medio de coeficientes distintos de cero para el subconjunto. El codificador 20 puede configurarse para recopilar estadísticas que indican con qué frecuencia los coeficientes en las posiciones dentro de cada subconjunto de unidades de datos de vídeo a nivel de hoja previamente codificadas incluyen coeficientes distintos de cero. Por ejemplo, el codificador 20 puede mantener uno o más contadores que cuentan, a medida que se codifican las unidades a nivel de hoja, con qué frecuencia los coeficientes dentro de cada subconjunto no son cero, y/o cuántos coeficientes de cada subconjunto no son cero. De acuerdo con este ejemplo, cuando el codificador 20 codifica una nueva unidad de datos de vídeo a nivel de hoja, el codificador 20 puede acceder a dicho contador para determinar el número medio de coeficientes distintos de cero para cada subconjunto. En algunos ejemplos, el contador para cada subconjunto puede reiniciarse (por ejemplo, inicializarse) periódicamente, por ejemplo con cada trama o fragmento de vídeo.

55 **[0101]** Como también se muestra en la FIG. 7, el codificador 20 compara el número medio determinado de coeficientes distintos de cero con un umbral $thAvgCoeff$ (703). Como también se muestra en la FIG. 7, si el número medio determinado de coeficientes distintos de cero es menor que el umbral $thAvgCoeff$, el codificador 20 genera el elemento sintáctico asociado con el subconjunto (704). Sin embargo, si el número medio determinado de coeficientes distintos de cero es mayor o igual que el umbral, el codificador 20 no genera el elemento sintáctico asociado con el subconjunto (705).

65 **[0102]** Como se describió anteriormente, el codificador 20 puede generar un flujo de bits codificado por entropía que incluye al menos un elemento sintáctico que indica si un subconjunto de coeficientes de transformada incluye algún coeficiente distinto de cero. El codificador 20 puede determinar si generar el elemento sintáctico basándose en una

serie de coeficientes potenciales distintos de cero del subconjunto como se muestra en la FIG. 6, o basado en un número medio de coeficientes distintos de cero, como se muestra en la FIG. 7. Un descodificador 30 puede leer el flujo de bits codificado por entropía, incluido el al menos un elemento sintáctico, y usar el al menos un elemento sintáctico para descodificar el flujo de bits codificado por entropía.

[0103] La FIG. 8 es un diagrama de flujo que ilustra un ejemplo de un procedimiento que puede ser realizado por un descodificador consistente con uno o más aspectos de esta divulgación. El procedimiento de la FIG. 8 se describe como realizado por el descodificador 30 representado en la FIG. 3; sin embargo, se puede usar cualquier dispositivo para realizar el procedimiento de la FIG. 8.

[0104] Como se muestra en la FIG. 8, el descodificador 30 puede dividir una unidad de datos de vídeo a nivel de hoja en una pluralidad de subconjuntos, cada uno de los cuales incluye múltiples coeficientes de transformada (801). Como un ejemplo, la pluralidad de subconjuntos puede comprender subconjuntos de forma rectangular como se representa en el ejemplo de la FIG. 4, o cualquier otra forma geométrica o patrón. La unidad de datos de vídeo a nivel de hoja puede comprender una unidad de codificación a nivel más bajo de un estándar de codificación de vídeo, tal como un nodo hijo no dividido de una estructura de árbol cuádruple como se describió anteriormente. La unidad de datos a nivel de hoja está representada por un flujo de bits codificado por entropía que incluye al menos un elemento sintáctico que indica si un subconjunto de la pluralidad de subconjuntos incluye coeficientes distintos de cero.

[0105] Como también se muestra en la FIG. 8, para cada uno de los subconjuntos, el descodificador 30 determina si el flujo de bits codificado por entropía incluye un elemento sintáctico asociado con el subconjunto (802). Como también se muestra en la FIG. 8, si el subconjunto no incluye dicho elemento sintáctico asociado, el descodificador 30 descodifica el subconjunto (por ejemplo, los coeficientes de transformada del subconjunto) (803). Como también se muestra en la FIG. 8, si el subconjunto incluye dicho elemento sintáctico asociado, el descodificador 30 puede usar el elemento sintáctico asociado para determinar si descodifica el subconjunto (804). Por ejemplo, si el elemento sintáctico tiene un primer valor (por ejemplo, un uno (1)), el descodificador 30 descodifica el subconjunto. Sin embargo, si el elemento sintáctico tiene un segundo valor (por ejemplo, un cero (0)), el descodificador 30 no descodifica el subconjunto.

[0106] En uno o más ejemplos, las funciones descritas en el presente documento pueden implementarse al menos parcialmente en hardware, como componentes de hardware específicos o un procesador. Más generalmente, las técnicas pueden implementarse en hardware, procesadores, software, firmware o cualquier combinación de los mismos. Si se implementan en software, las funciones se pueden almacenar en, o transmitir por, un medio legible por ordenador, como una o más instrucciones o código, y ejecutarse mediante una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible tal como medios de almacenamiento de datos, o medios de comunicación que incluyen cualquier medio que facilite la transferencia de un programa informático desde un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador pueden corresponder, en general, a (1) medios de almacenamiento tangibles legibles por ordenador que sean no transitorios o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que se pueda acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en esta divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

[0107] A modo de ejemplo, y no de limitación, tales medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento de disco óptico, almacenamiento de disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que pueda usarse para almacenar código de programa deseado en forma de instrucciones o estructuras de datos y al que pueda accederse mediante un ordenador. Además, cualquier conexión se denomina correctamente un medio legible por ordenador, es decir, un medio de transmisión legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, un servidor u otro origen remoto usando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas se incluyen en la definición de medio. Sin embargo, debería entenderse que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales ni otros medios transitorios, sino que, en cambio, se orientan a medios de almacenamiento tangibles no transitorios. Los discos, como se usan en el presente documento, incluyen el disco compacto (CD), el disco láser, el disco óptico, el disco versátil digital (DVD), el disco flexible y el disco Blu-ray, donde algunos discos reproducen habitualmente los datos magnéticamente, mientras que otros discos reproducen los datos ópticamente con láseres. Las combinaciones de lo anterior se deberían incluir también dentro del alcance de los medios legibles por ordenador.

[0108] Las instrucciones pueden ser ejecutadas por uno o más procesadores, tales como uno o más unidades centrales de procesamiento (CPU), procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), formaciones lógicas programables in situ (FPGA) u otros circuitos lógicos equivalentes, integrados o discretos. En consecuencia, el término "procesador", como se usa en el

presente documento, se puede referir a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento se puede proporcionar en módulos de hardware y/o software dedicados configurados para la codificación y la decodificación, o incorporados en un códec combinado. Además, las técnicas se podrían implementar totalmente en uno o más circuitos o elementos lógicos.

[0109] Las técnicas de esta divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluidos un equipo manual inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). Diversos componentes, módulos o unidades se describen en esta divulgación para enfatizar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no requieren necesariamente su realización mediante diferentes unidades de hardware. En cambio, como se ha descrito anteriormente, diversas unidades se pueden combinar en una unidad de hardware de códec o proporcionarse mediante un grupo de unidades de hardware interoperativas, incluidos uno o más procesadores, como se ha descrito anteriormente, conjuntamente con software y/o firmware adecuados.

[0110] Se han descrito diversos ejemplos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.

REIVINDICACIONES

1. Un procedimiento de codificación de una unidad de datos de vídeo, que comprende:
- 5 dividir (501, 601, 701) un bloque (410) de coeficientes de transformada cuantificados (412) en una pluralidad de subbloques cuadrados más pequeños (420) de coeficientes de transformada cuantificados (412); **caracterizado por**
- 10 determinar, para un subbloque de la pluralidad de subbloques de coeficientes de transformada cuantificados, si se debe indicar (502) un elemento sintáctico que indique si el subbloque incluye o no cualquier coeficiente de transformada cuantificado distinto de cero.
- 15 2. El procedimiento según la reivindicación 1, en el que determinar, para el subbloque, si se debe señalar o no el elemento sintáctico comprende determinar si la señal del elemento sintáctico mejorará la eficiencia de codificación.
- 20 3. El procedimiento según la reivindicación 1, en el que determinar si se debe señalar o no el elemento sintáctico comprende comparar una cantidad de coeficientes de transformada cuantificados potenciales distintos de cero del subbloque con al menos un umbral; y
- 25 si el número de coeficientes de transformada cuantificados potenciales distintos de cero del subbloque es mayor que el umbral, indica el elemento sintáctico.
- 30 4. El procedimiento según la reivindicación 3, que comprende además:
- 35 determinar (603) el número de coeficientes de transformada cuantificados no nulos potenciales basándose en cuántos (602) coeficientes del subbloque tienen una posición después de un último coeficiente de transformada cuantificado no nulo del bloque (410) de coeficientes de transformada cuantificados.
- 40 5. El procedimiento según la reivindicación 1, en el que la determinación de si señalar el elemento sintáctico comprende:
- 45 determinar (702), para una pluralidad de bloques previamente codificados (410) de coeficientes de transformada cuantificados de datos de vídeo, un número medio de coeficientes de transformada cuantificados distintos de cero incluidos en el subbloque;
- 50 comparar (703) el número medio de coeficientes de transformada cuantificados distintos de cero incluidos en el subbloque con un umbral; y
- 55 si el número medio de coeficientes de transformada cuantificados distintos de cero es menor que el umbral, señalar el elemento sintáctico.
- 60 6. El procedimiento según la reivindicación 1, que comprende además:
- 65 determinar, para cada subbloque de la pluralidad de subbloques de coeficientes de transformada cuantificados, si se debe señalar o no el elemento sintáctico.
7. Un procedimiento de descodificación de una unidad de datos de vídeo, que comprende:
- 80 dividir (801) un bloque (410) de coeficientes de transformada cuantificados (412) en una pluralidad de subbloques cuadrados más pequeños (420) de coeficientes de transformada cuantificados; **caracterizado por**
- 85 determinar (802), para un subbloque de la pluralidad de subbloques de coeficientes de transformada cuantificados, si el subbloque incluye un elemento sintáctico asociado que indica si el subbloque incluye o no coeficientes de transformada cuantificados distintos de cero; y
- 90 determinar (803, 804), basándose en el elemento sintáctico, si descodificar o no el subbloque.
- 95 8. El procedimiento según la reivindicación 7, que comprende además:
- 100 si el subbloque no incluye el elemento sintáctico, descodificar el subbloque.
- 105 9. El procedimiento según la reivindicación 8, que comprende además:

si el subbloque de datos incluye el elemento sintáctico, determinar si descodificar el subbloque basándose en la indicación.

- 5 **10.** El procedimiento según la reivindicación 9, que comprende además:
- si el elemento sintáctico asociado con el subbloque tiene un primer valor, descodificar el subbloque; y
- si el elemento sintáctico tiene un segundo valor diferente al primer valor, no descodificar el subbloque.
- 10 **11.** El procedimiento según la reivindicación 7, que comprende además:
- determinar, para cada subbloque de la pluralidad de subbloques de coeficientes de transformada cuantificados, si el subbloque incluye un elemento sintáctico asociado que indica si el subbloque incluye o no cualquier coeficiente de transformada cuantificado distinto de cero; y
- 15 determinar, basándose en el elemento sintáctico, si descodificar o no el subbloque.
- 12.** Un dispositivo que comprende un codificador de vídeo, en el que el codificador de vídeo comprende:
- 20 medios para dividir (501, 601, 701) un bloque (410) de coeficientes de transformada cuantificados en una pluralidad de subbloques cuadrados más pequeños (420) de transformada cuantificada; con el dispositivo que se **caracteriza por**
- medios para determinar, para un subbloque de la pluralidad de subbloques de coeficientes de transformada cuantificados, si indicar o no (502) un elemento sintáctico que indique si el subbloque incluye o no cualquier coeficiente de transformada cuantificado distinto de cero.
- 25 **13.** Un dispositivo que comprende un descodificador de vídeo, en el que el descodificador de vídeo comprende:
- 30 medios para dividir (801) un bloque de coeficientes de transformada cuantificados en una pluralidad de subbloques cuadrados más pequeños de coeficiente de transformada cuantificado; con el dispositivo que se **caracteriza por**
- medios para determinar (802), para un subbloque de la pluralidad de subbloques de coeficientes de transformada cuantificados, si el subbloque incluye un elemento sintáctico asociado que indica si el subbloque incluye o no cualquier coeficiente de transformada cuantificada distinto de cero; y
- 35 medios para determinar (803, 804), basándose en el elemento sintáctico, si descodificar o no el subbloque.
- 40 **14.** Un medio de almacenamiento legible por ordenador que almacena instrucciones que, cuando son ejecutadas, hacen que un dispositivo informático realice un procedimiento de acuerdo con cualquiera de las reivindicaciones 1 a 11.

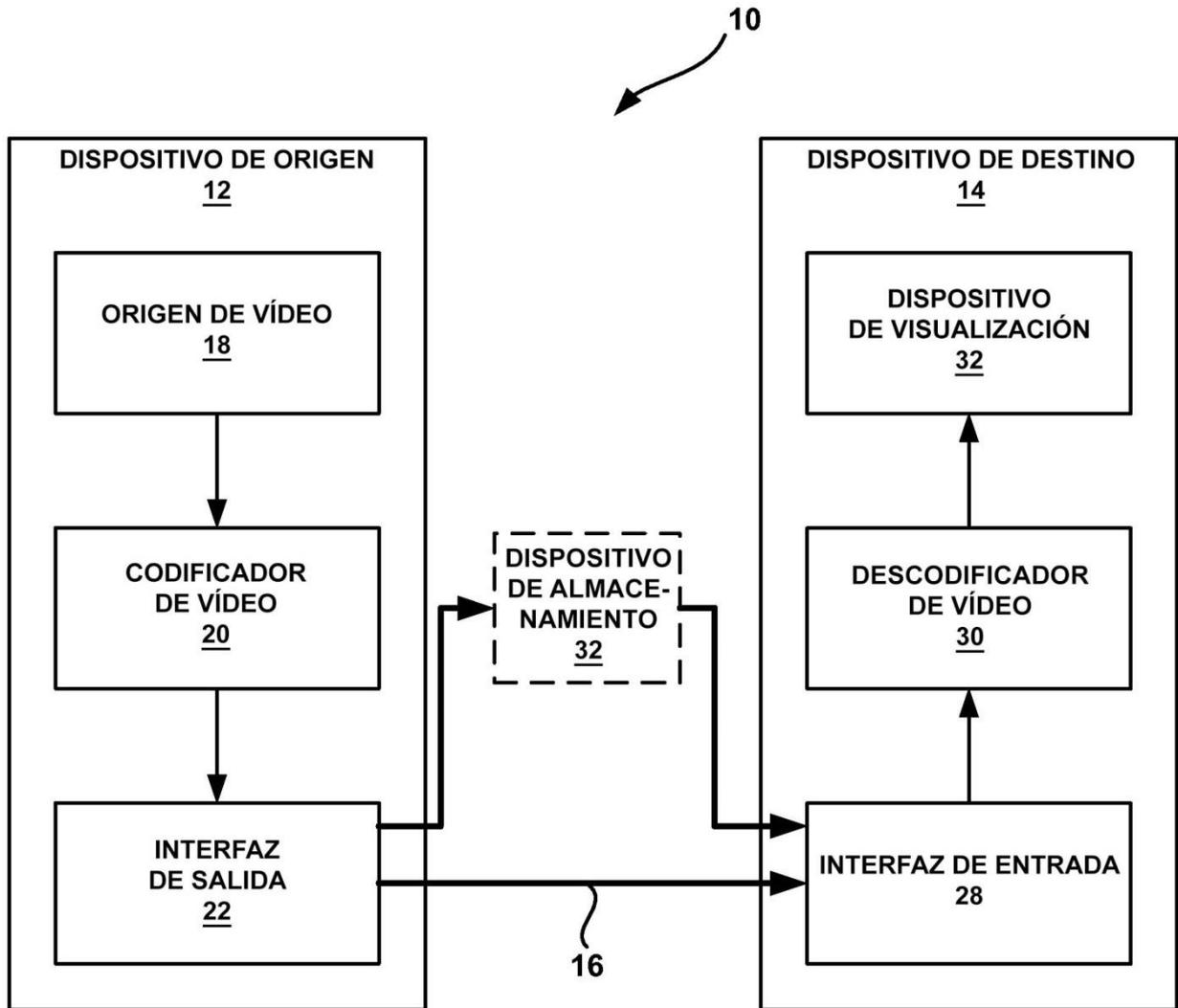


FIG. 1

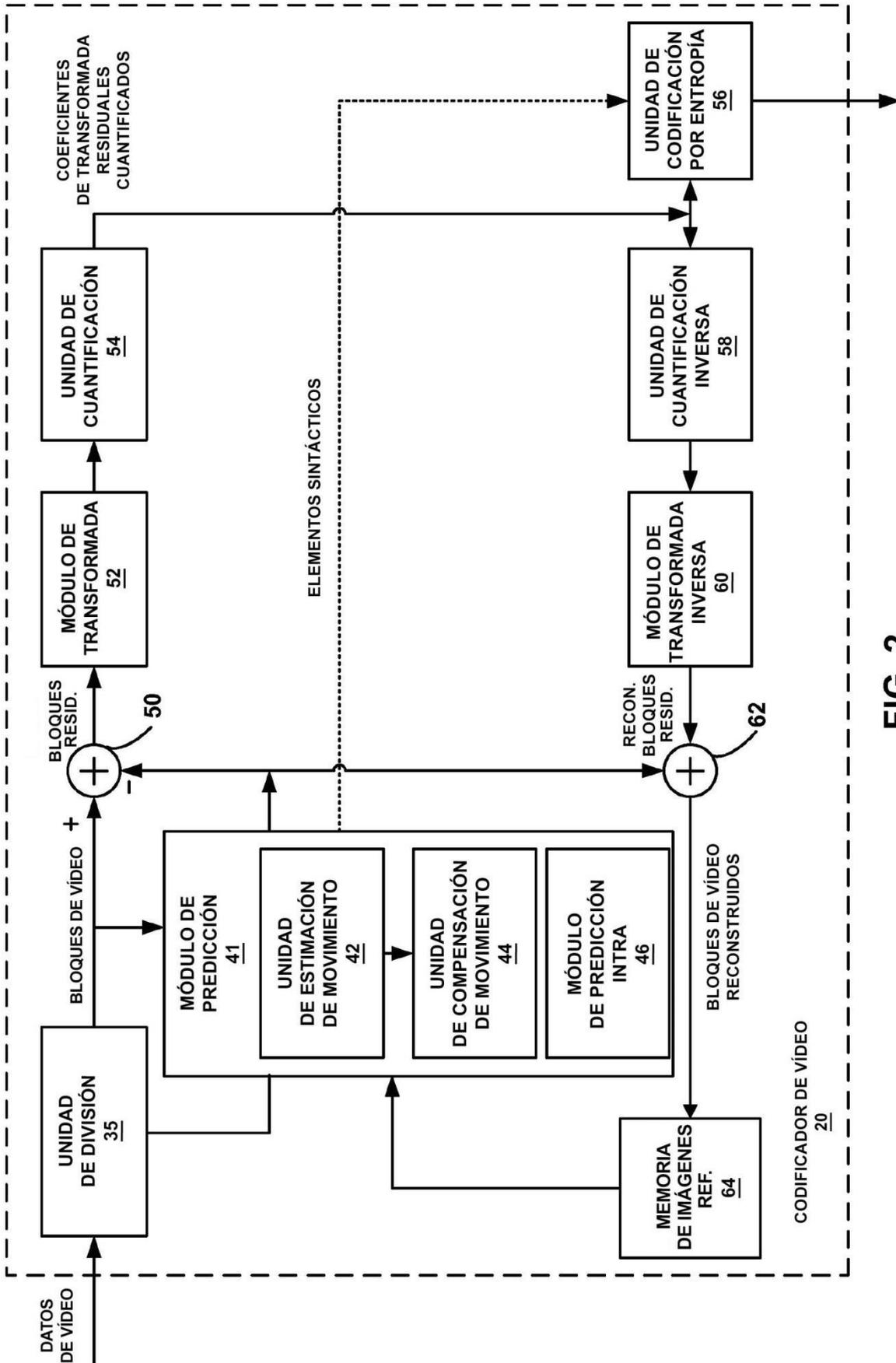


FIG. 2

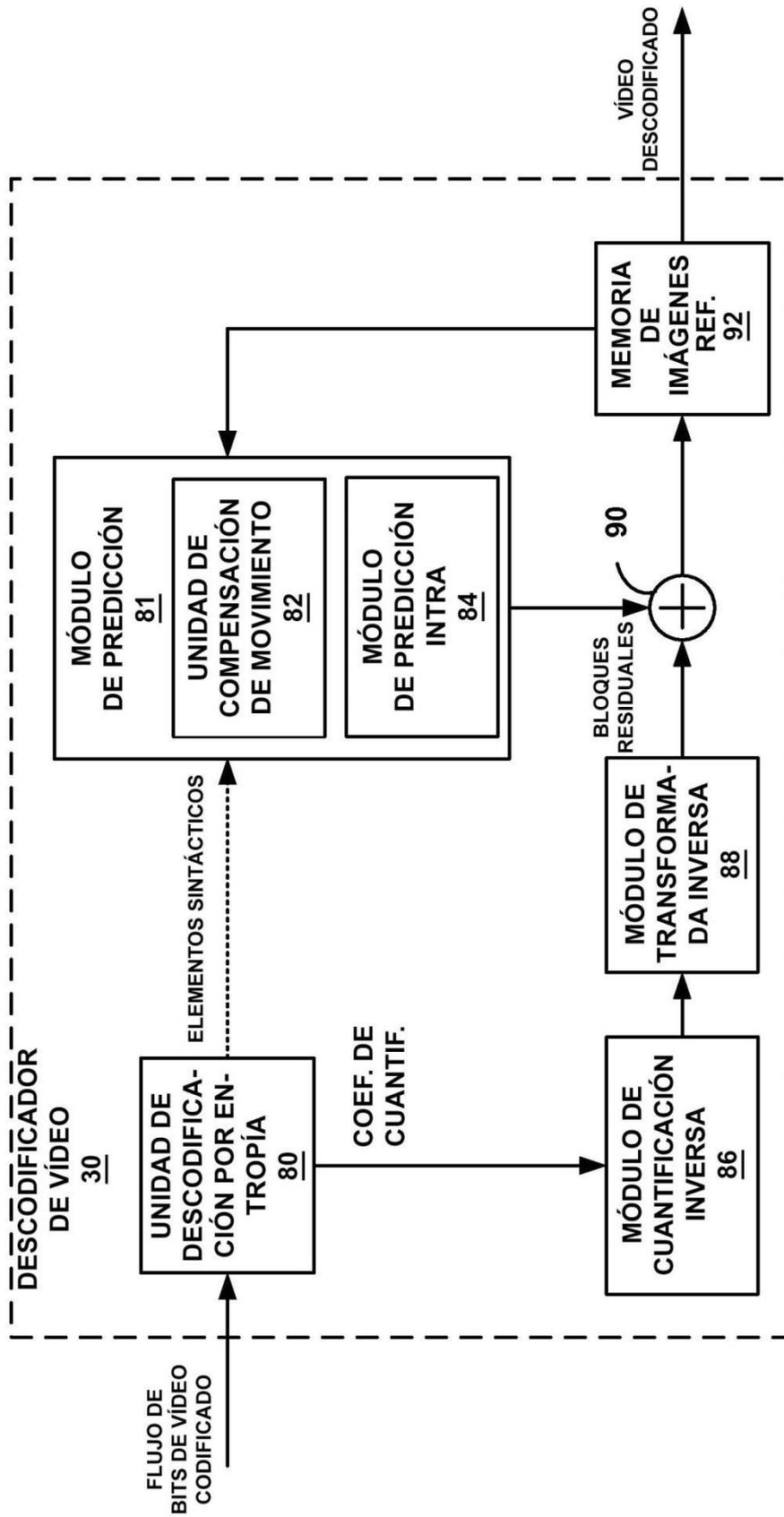


FIG. 3

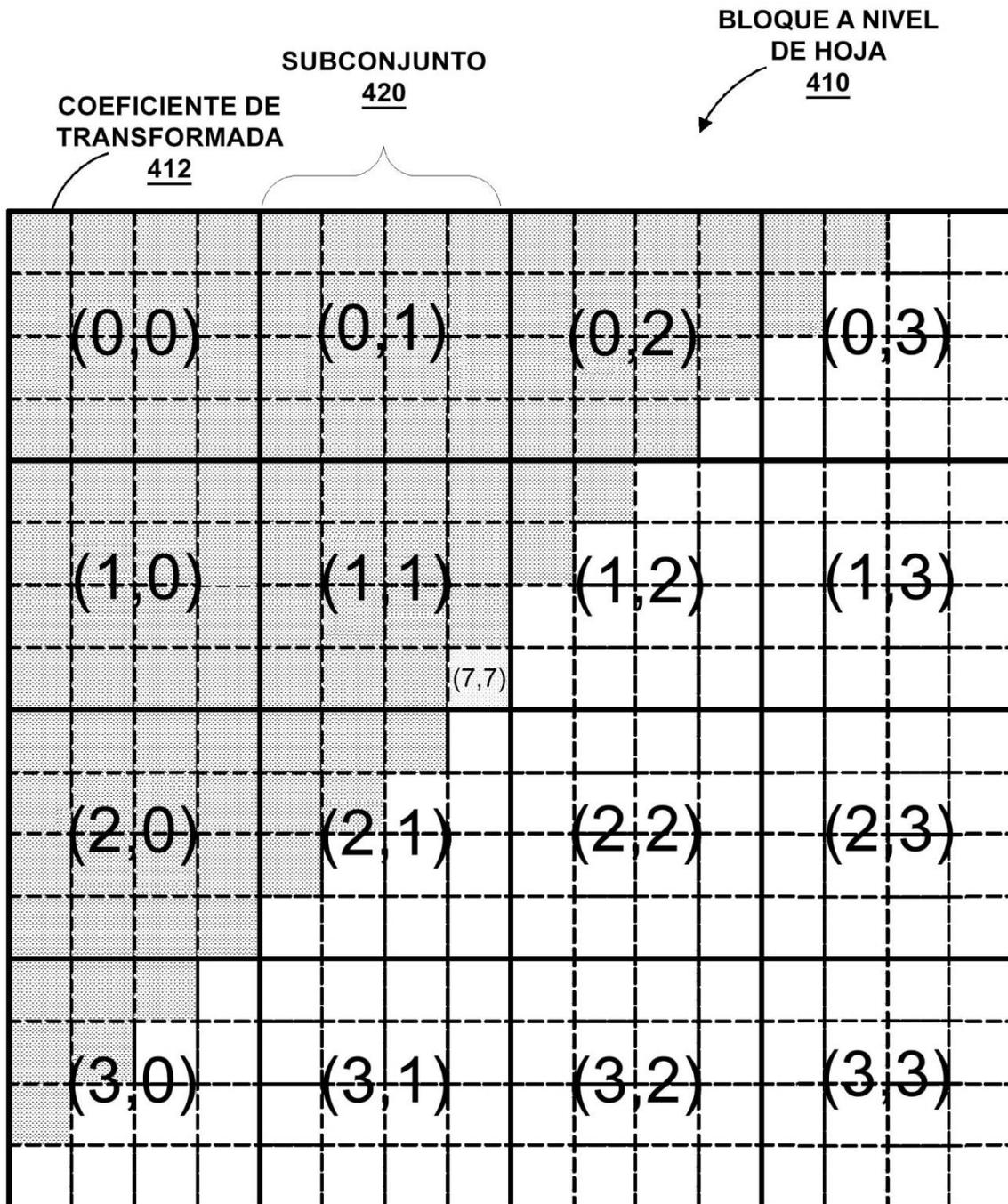


FIG. 4

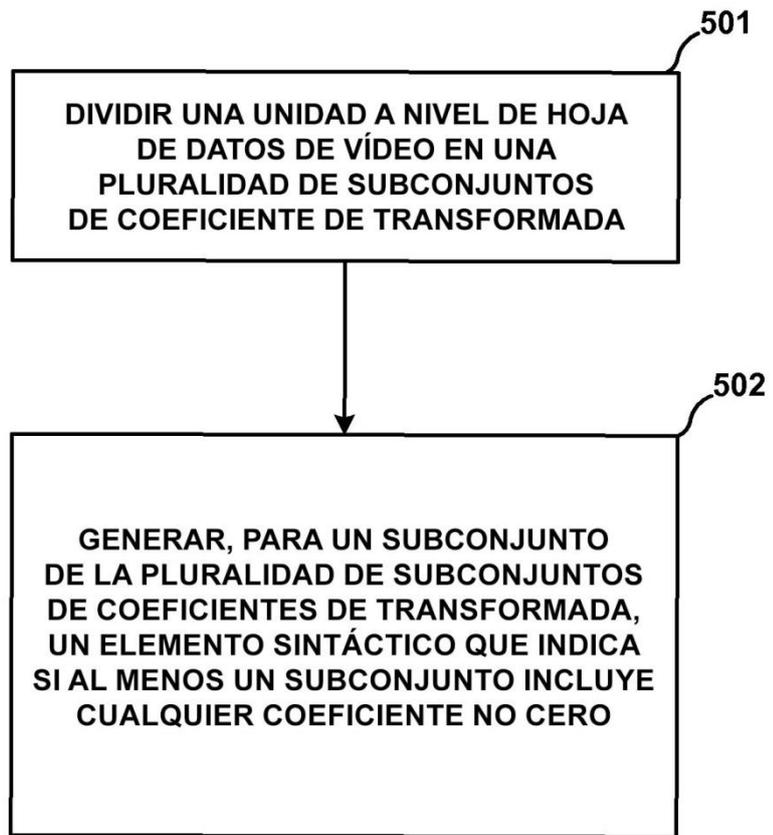


FIG. 5

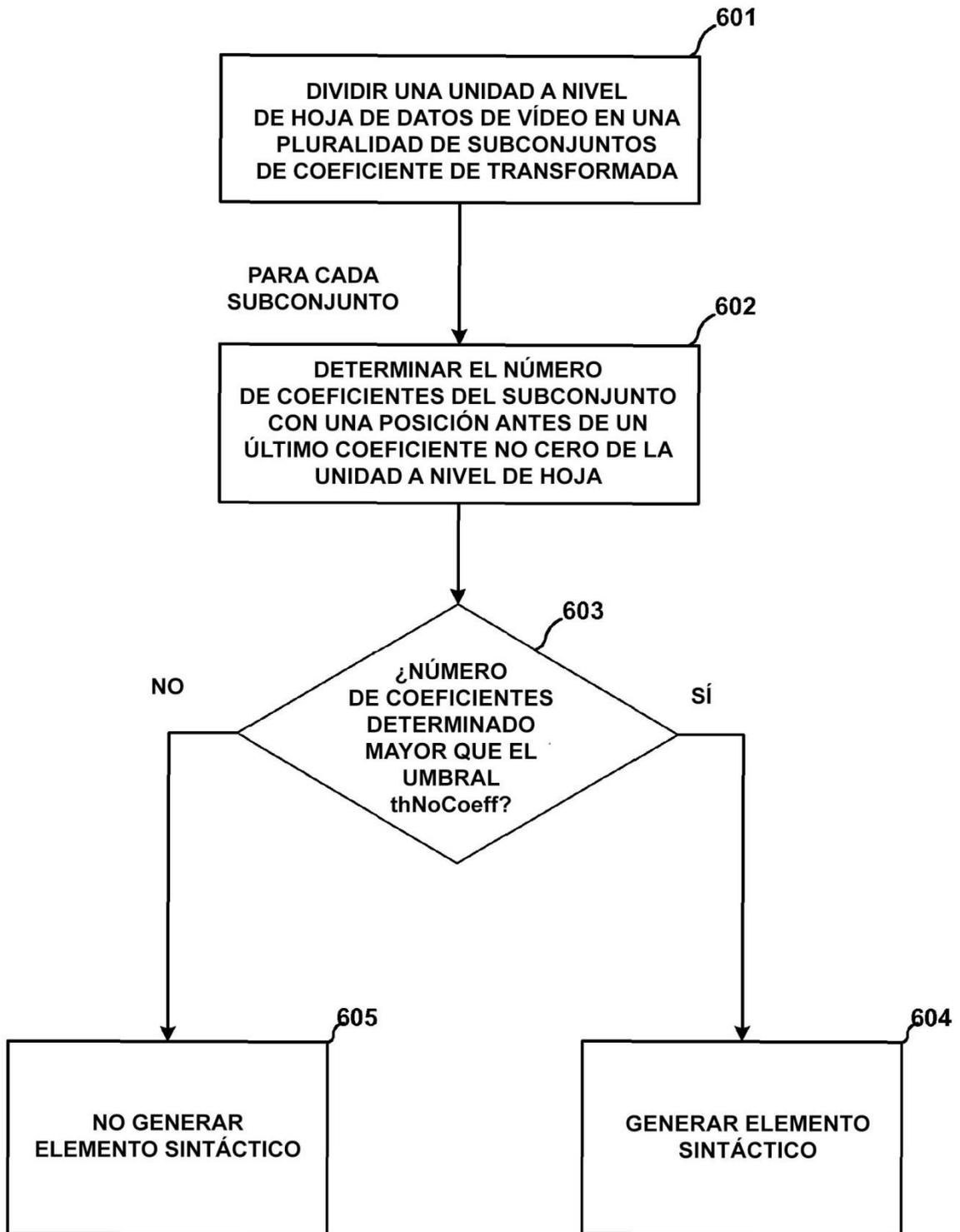


FIG. 6

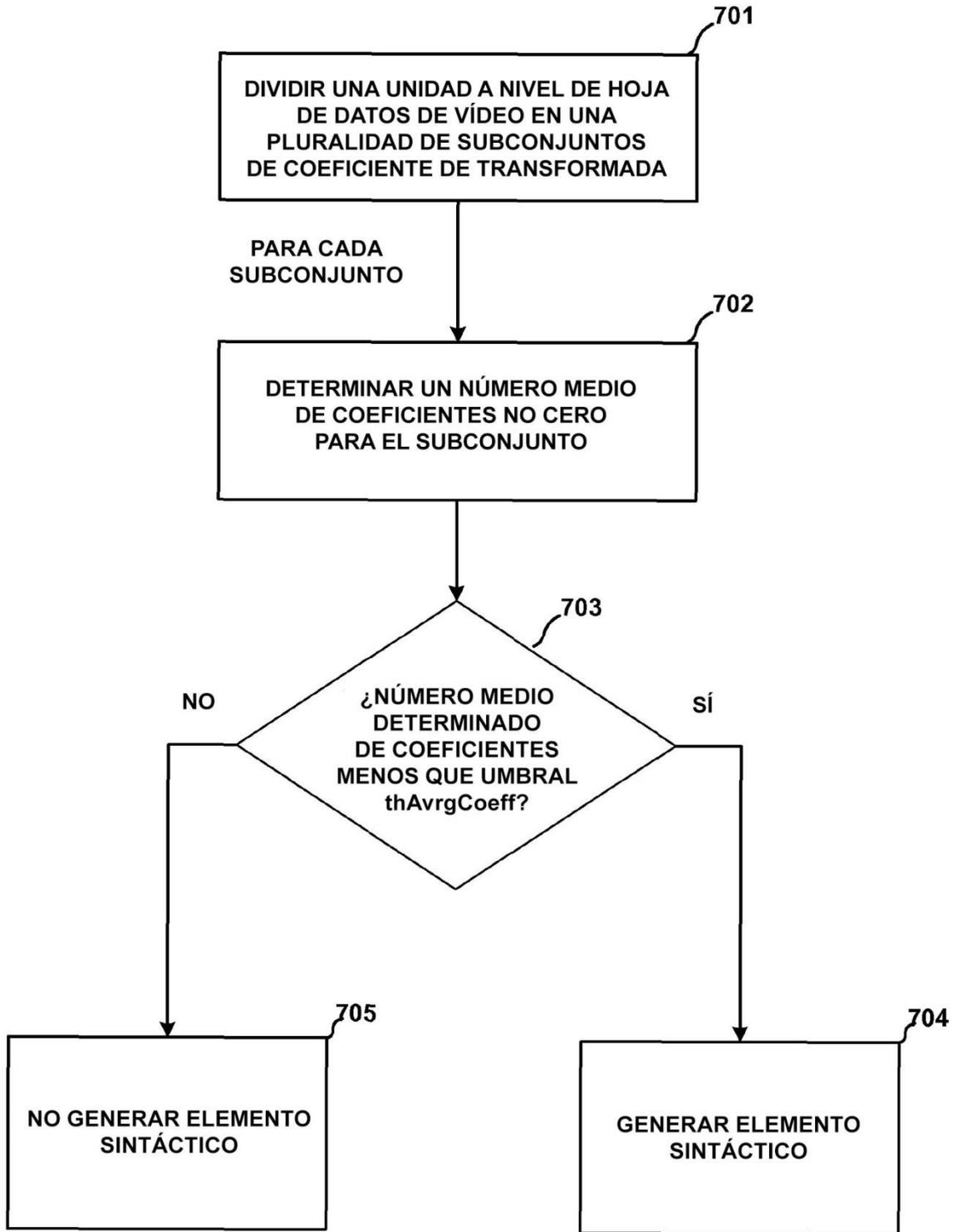


FIG. 7

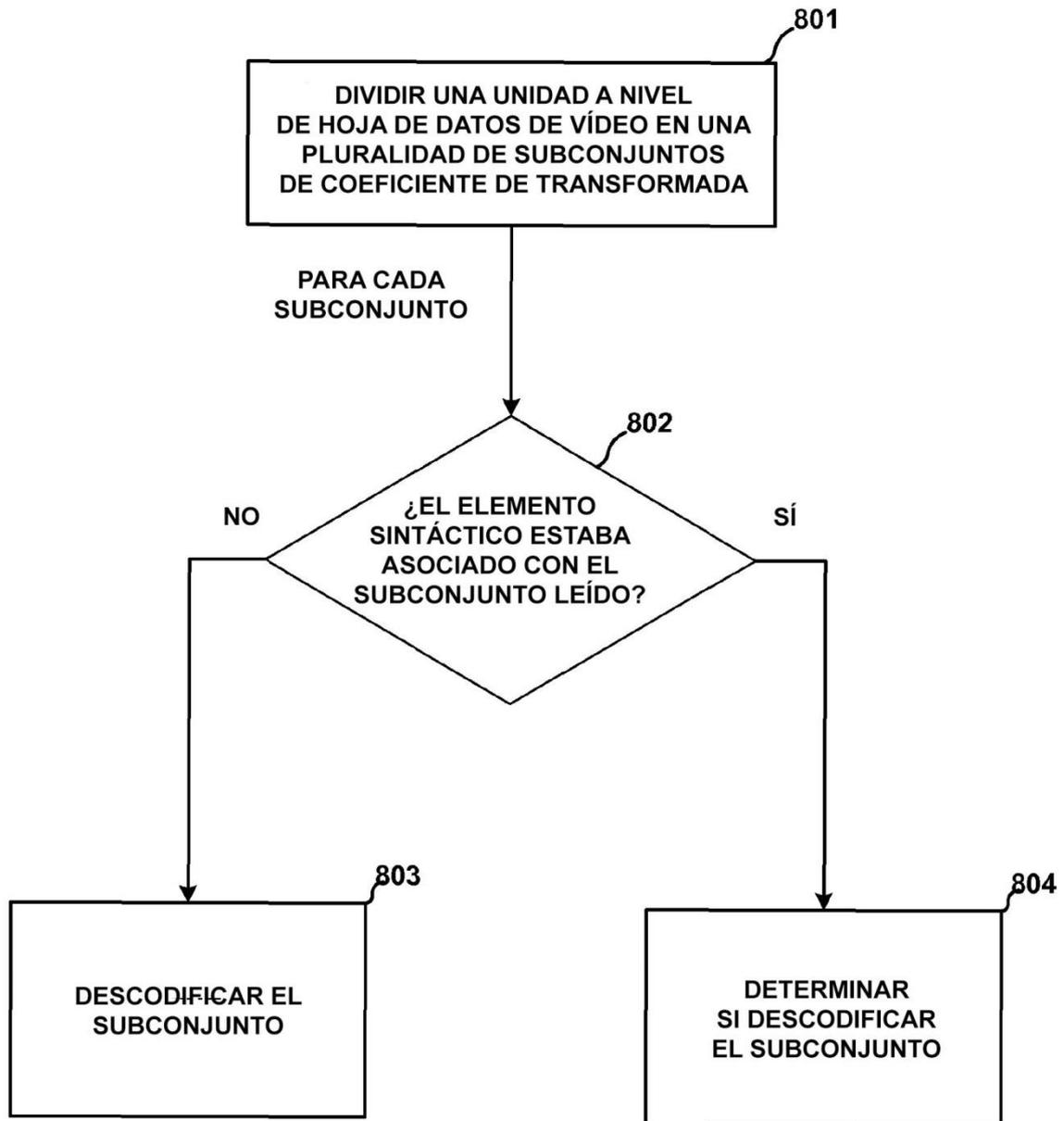


FIG. 8