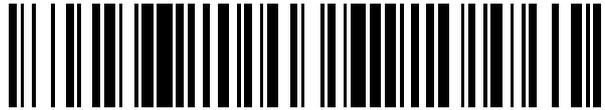


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 546 678**

51 Int. Cl.:

H04N 19/52 (2014.01)
H04N 19/70 (2014.01)
H04N 19/513 (2014.01)
H04N 19/13 (2014.01)
H04N 19/91 (2014.01)
H04N 19/61 (2014.01)
H04N 19/523 (2014.01)
H04N 19/517 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **29.06.2012 E 12733835 (8)**

97 Fecha y número de publicación de la concesión europea: **15.07.2015 EP 2727353**

54 Título: **Codificación de vídeo usando resolución adaptativa de vectores de movimiento**

30 Prioridad:

01.07.2011 US 201161504150 P
01.11.2011 US 201161554398 P
28.06.2012 US 201213536218

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
25.09.2015

73 Titular/es:

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

72 Inventor/es:

CHIEN, WEI-JUNG;
CHEN, PEISONG y
KARCZEWICZ, MARTA

74 Agente/Representante:

CARPINTERO LÓPEZ, Mario

ES 2 546 678 T3

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

DESCRIPCIÓN

Codificación de vídeo usando resolución adaptativa de vectores de movimiento

Campo técnico

5 La presente divulgación se refiere a la codificación de vídeo y, más específicamente, a la codificación de datos de vídeo para la compensación de movimiento en la codificación de vídeo.

Antecedentes

10 Las capacidades del vídeo digital pueden ser incorporadas en una amplia gama de dispositivos, incluyendo los televisores digitales, los sistemas digitales de difusión directa, los sistemas de difusión inalámbrica, los asistentes digitales personales (PDA), los ordenadores portátiles o de sobremesa, las cámaras digitales, los dispositivos de grabación digital, los reproductores de medios digitales, los dispositivos de juegos de vídeo, las consolas de juegos de vídeo, los radioteléfonos celulares o por satélite, los dispositivos de teleconferencia de vídeo y similares. Los dispositivos de vídeo digital implementan técnicas de compresión de vídeo, tales como las descritas en los estándares definidos por MPEG-2, MPEG-4, ITU-T H.263 o ITU-T H.264 / MPEG-4, Parte 10, Codificación de Vídeo Avanzada (AVC) y las extensiones de tales estándares, para transmitir y recibir más eficazmente información de vídeo digital.

15 Las técnicas de compresión de vídeo realizan la predicción espacial y / o la predicción temporal para reducir o eliminar la redundancia inherente en las secuencias de vídeo. Para la codificación basada en bloques, una trama o tajada de vídeo puede ser dividida en macro-bloques. Cada macro-bloque puede ser dividido adicionalmente. Los macro-bloques en una trama o tajada intra-codificada (I) son codificados usando la predicción espacial con respecto a los macro-bloques vecinos. Los macro-bloques en una trama o tajada inter-codificada (P o B) pueden usar la predicción espacial con respecto a los macro-bloques vecinos en la misma trama o tajada, o la predicción temporal con respecto a otras tramas de referencia.

20 El artículo "Codificación aritmética binaria adaptativa basada en el contexto en el estándar de compresión de vídeo H.264 / AVC" de Marpe et al, publicado en las Transacciones del IEEE sobre Circuitos y Sistemas, vol. 13, nº 7, páginas 620 a 636, del 1 de julio de 2003, describe una técnica para codificar el bit de signo para un componente x o y, junto con un valor representativo de la magnitud del valor de diferencia de vector de movimiento para ese componente x o y.

25 El documento US 2005 / 038837 describe un procedimiento y aparato para la binarización y la codificación aritmética de un valor de datos.

30 El artículo "CE12: Resolución adaptativa de vectores de movimiento de Qualcomm", 4. Reunión JCT-VC; 95. Reunión de MPEG, nº JCTVC-D394, del 16 de enero de 2011, describe un procedimiento de resolución adaptativa de vectores de movimiento dentro del ámbito del experimento central 12 que fue configurado en la 3ª reunión de JCT-VC.

Sumario

La invención está definida en las reivindicaciones a las cuales se dirige ahora la referencia.

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

Breve descripción de los dibujos

35 La FIG. 1 es un diagrama de bloques que ilustra un ejemplo de un sistema de codificación y descodificación de vídeo congruente con las técnicas de esta divulgación.

La FIG. 2 es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo congruente con las técnicas de esta divulgación.

40 La FIG. 3 es un diagrama de bloques que ilustra un ejemplo de un descodificador de vídeo congruente con las técnicas de esta divulgación.

La FIG. 4 es un diagrama de flujo que ilustra un procedimiento ejemplar para codificar conjuntamente los componentes x e y de un valor de diferencia de vector de movimiento y para codificar por entropía un bloque actual.

45 La FIG. 5 es un diagrama de flujo que ilustra un procedimiento ejemplar de señalización por separado de un valor de diferencia de vector de movimiento de acuerdo al modelo de prueba HEVC 3.0 (HM 3.0).

La FIG. 6 es un diagrama de flujo que ilustra un procedimiento ejemplar para codificar conjuntamente los componentes x e y de un valor de diferencia de vector de movimiento.

La FIG. 7 es un diagrama de flujo que ilustra un procedimiento ejemplar para codificar conjuntamente los componentes x e y de un valor de diferencia de vector de movimiento y para codificar la resolución de los componentes x e y de un valor de diferencia de vector de movimiento.

5 La FIG. 8 es un diagrama de flujo que ilustra un procedimiento ejemplar para descodificar conjuntamente los componentes x e y de un valor de diferencia de vector de movimiento y para descodificar datos codificados por entropía, para producir un bloque actual.

La Fig. 9 es un diagrama de flujo que ilustra un procedimiento ejemplar para descodificar conjuntamente los componentes x e y de un valor de diferencia de vector de movimiento.

Descripción detallada

10 En general, esta divulgación describe técnicas para codificar datos para valores de diferencia de vectores de movimiento durante la codificación de vídeo. Los datos de vídeo incluyen una secuencia de tramas (o imágenes) reproducidas en rápida sucesión para simular el movimiento. Cada trama de datos de vídeo puede ser dividida en bloques. Durante la codificación de vídeo y, más específicamente, la inter-predicción, un dispositivo tal como un codificador de vídeo o un
15 descodificador de vídeo puede predecir los valores de píxeles de un bloque. El codificador o descodificador de vídeo puede basar las predicciones para el bloque sobre los valores de píxel de un bloque procedente de otra trama, o a partir de los valores de píxel de un bloque vecino.

Para la intra-predicción, un codificador de vídeo puede indicar el desplazamiento de un bloque de referencia usando un vector de movimiento. El vector de movimiento puede tener un componente x y un componente y. Los componentes x e y del vector de movimiento pueden indicar un desplazamiento con precisión de sub-píxel, tal como una precisión de medio
20 píxel, una precisión de un cuarto de píxel o una precisión de un octavo de píxel. Para lograr la precisión sub-píxel, un codificador o descodificador de vídeo puede usar una técnica, tal como la interpolación, para determinar los valores de sub-píxel en las ubicaciones indicadas por el vector de movimiento. Después de determinar los componentes x e y del vector de movimiento, un codificador de vídeo puede calcular valores de diferencia de vectores de movimiento (MVD) para los componentes del vector de movimiento, con respecto a un predictor de vector de movimiento.

25 Esta divulgación proporciona, en general, técnicas para codificar conjuntamente valores MVD de movimiento. En esta divulgación, la codificación conjunta puede referirse a técnicas de codificación en las cuales las informaciones referidas a los componentes x e y de un valor de diferencia de vector de movimiento están intercaladas, a diferencia de la codificación de toda la información referida a uno de los componentes x o y antes de codificar cualquier información referida al segundo componente del valor de diferencia de vector de movimiento. La codificación conjunta también puede referirse al
30 uso de un valor para representar valores para ambos componentes x e y. Además, esta divulgación también describe técnicas para codificar diferencias de vectores de movimiento, que pueden tener diversas precisiones sub-píxel, p. ej., precisión de un cuarto de píxel o precisión de un octavo de píxel, y para indicar las precisiones de las diferencias de vectores de movimiento y los vectores de movimiento asociados a los vectores de movimiento.

Una secuencia de vídeo incluye una o más tramas o imágenes. Cada una de las imágenes puede ser dividida en uno o
35 más bloques, cada uno de los cuales puede ser codificado individualmente. Están en marcha esfuerzos para producir un nuevo estándar de codificación, actualmente mencionado como codificación de vídeo de alta eficacia (HEVC), y algunas veces mencionado como ITU H.265. Este estándar inminente se refiere a una unidad de codificación (CU) como un bloque específico de píxeles que incluye datos de luminancia y datos de crominancia, donde los datos de luminancia tienen una resolución de 2Nx2N y los datos de crominancia tienen una resolución de NxN. Una unidad de codificación
40 puede ser dividida en cuatro unidades de sub-codificación no solapadas, cuadradas, de igual tamaño.

Cada una de las unidades de sub-codificación puede también ser dividida de esta manera en unidades adicionales de sub-codificación. Una unidad de codificación que no está dividida en unidades de sub-codificación se menciona como una unidad de codificación de nodo hoja. Las unidades de codificación de nodo hoja pueden incluir unidades de predicción (PU) y unidades de transformación (TU), donde las PU representan datos de predicción y las TU representan datos
45 residuales, es decir, diferencias codificadas, píxel a píxel, entre los datos de predicción y los datos originales, no codificados, para los píxeles correspondientes a la TU. Como ejemplo, las PU pueden ser codificadas usando una modalidad de inter-predicción, en la cual un codificador de vídeo puede calcular un vector de movimiento para una PU usando un proceso de estimación de movimiento. El codificador de vídeo puede además señalar una modalidad de codificación para la PU y valores de diferencia de vectores de movimiento para el vector de movimiento calculado, según
50 lo descrito en esta divulgación.

Análogamente, un descodificador de vídeo puede usar información indicativa de una modalidad de predicción incluida en un flujo de bits codificado para formar datos de predicción para bloques codificados. Los datos pueden además incluir una precisión del vector de movimiento, así como una indicación de una posición de píxel fraccionaria a la cual apunta el vector de movimiento (por ejemplo, una posición de un octavo de píxel de una trama de referencia o una tajada de referencia).
55

Un dispositivo de codificación de vídeo, tal como un codificador de vídeo o un decodificador de vídeo, puede determinar un vector de movimiento predictivo para una unidad de codificación (tal como una trama, tajada o bloque). El dispositivo de codificación de vídeo puede usar componentes x e y del vector de movimiento predictivo (también mencionado como un predictor de vector de movimiento) para calcular valores de diferencia de vectores de movimiento de un vector de movimiento para el bloque actual. Un dispositivo de codificación de vídeo puede implementar las técnicas de esta divulgación para codificar conjuntamente valores de diferencia de vectores de movimiento para vectores de movimiento, que pueden tener precisión de sub-píxel.

Como se ha expuesto anteriormente, los datos de predicción de una PU para formar un bloque predictivo pueden estar basados en datos previamente codificados de CU espacialmente vecinas, o CU de tramas temporalmente vecinas que han sido codificadas previamente. Un codificador de vídeo puede calcular un vector de movimiento de una PU que indica la ubicación de un bloque predictivo para una CU correspondiente en una trama previamente codificada, temporalmente separada. El codificador de vídeo puede además calcular un valor de diferencia de vector de movimiento para el vector de movimiento, para codificar el vector de movimiento. El valor de diferencia de vector de movimiento puede corresponder, en general, a una diferencia entre el vector de movimiento calculado y un predictor de vector de movimiento. El vector de movimiento para el bloque actual puede tener un componente x (MV_x) para un desplazamiento horizontal, y un componente y (MV_y) para un desplazamiento vertical. El predictor del vector de movimiento puede tener un componente x de p_x y un componente y de p_y. Luego, el predictor del vector de movimiento puede ser determinado como $\langle MV_x - p_x, MV_y - p_y \rangle$.

Esta divulgación proporciona técnicas para codificar conjuntamente el componente x y el componente y de un valor de diferencia de vector de movimiento. En esta divulgación, la codificación conjunta puede referirse a técnicas de codificación en las cuales las informaciones referidas a los componentes x e y de un valor de diferencia de vector de movimiento están intercaladas. La información referida a la diferencia de vector de movimiento puede incluir un valor que representa si un componente de la diferencia de vector de movimiento es o no cero, uno o más valores que representan el valor del signo de los componentes x y / o los componentes y, y uno o más valores representativos del valor absoluto de los componentes x e y del valor de diferencia de vector de movimiento, como algunos ejemplos.

De acuerdo a las técnicas de esta divulgación, la información referida a uno o ambos componentes de un valor de diferencia de vector de movimiento puede ser codificada en forma alternativa, de modo que toda la información referida a un componente de la diferencia del vector de movimiento no deba necesariamente ser codificada antes de codificar información referida al otro componente de la diferencia del vector de movimiento. En otras palabras, un elemento del componente x, tal como si el componente x tiene o no una magnitud mayor que cero, puede ser intercalado con un correspondiente elemento del componente y. La codificación conjunta del componente x y del componente y de la diferencia del vector de movimiento contrasta con la codificación por separado de toda la información referida a uno entre el componente x y el componente y de un valor de diferencia de vector de movimiento, p. ej., según la convención en la norma ITU-T H.264. La codificación conjunta del componente x y del componente y de los valores de diferencia de vectores de movimiento puede reducir la complejidad de un flujo de bits de codificación de vídeo, lo que puede dar como resultado una menor velocidad de bits para el vídeo codificado, así como otras ventajas potenciales descritas en la presente memoria.

En la codificación de vídeo de acuerdo al estándar convencional ITU-T H.264, los vectores de movimiento pueden tener precisión de cuarto de píxel (es decir, un cuarto de píxel). En algunos casos, la precisión de octavo de píxel (es decir, un octavo de píxel) puede proporcionar ciertas ventajas. El Modelo de Prueba de Codificación de Vídeo de Alta Eficacia (HM) tiene la capacidad de formar unidades de predicción usando vectores de movimiento que tienen precisión de cuarto de píxel. En particular, el HM proporciona la capacidad de seleccionar adaptativamente precisión de cuarto de píxel o bien precisión de octavo de píxel. De esta manera, los vectores de movimiento pueden tener precisión adaptativa, también mencionada como resolución de vector de movimiento. Esta precisión de píxel fraccionaria, o sub-entera, puede ser usada para definir un vector de movimiento con respecto a un bloque de valores de píxeles interpolados en la trama previamente codificada.

Esta divulgación también proporciona técnicas para seleccionar casos particulares en los cuales pueden usarse vectores de movimiento de precisión de un cuarto o un octavo de píxel, y en cuanto a cómo puede ser señalizada la precisión del vector de movimiento para un vector de movimiento específico. En particular, las técnicas de esta divulgación están orientadas a seleccionar una resolución de vector de movimiento, y señalar la resolución usando un indicador de resolución de movimiento en algunos casos. Esta divulgación también proporciona técnicas para señalar conjuntamente los componentes x e y de un vector de movimiento específico.

Esta divulgación describe varias técnicas referidas a la codificación de vectores de movimiento, tales como la codificación conjunta de valores de diferencia de vector de movimiento y la señalización de la precisión, amplitud y signo de uno o más valores de diferencia de vectores de movimiento. Las técnicas de esta divulgación pueden ser realizadas durante un proceso de codificación realizado por un dispositivo de codificación de vídeo, tal como un codificador de vídeo o un decodificador de vídeo. En esta divulgación, el término "codificación" se refiere a la codificación que ocurre en el

codificador o a la decodificación que ocurre en el descodificador. De manera similar, el término codificador se refiere a un codificador, a un descodificador o a un codificador / descodificador (CODEC) combinado. Los términos codificador, descodificador y CODEC se refieren todos a máquinas específicas diseñadas para la codificación (codificación y / o decodificación) de datos de vídeo congruentes con esta divulgación.

5 La FIG. 1 es un diagrama de bloques que ilustra un sistema ejemplar de codificación y decodificación de vídeo 10 que puede utilizar técnicas para señalar conjuntamente vectores de movimiento. Los vectores de movimiento pueden tener precisión de sub-píxel. Según se muestra en la FIG. 1, el sistema 10 incluye un dispositivo de origen 12 que transmite vídeo codificado a un dispositivo de destino 14 mediante un canal de comunicación 16. El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender cualquiera entre una amplia gama de dispositivos. En algunos casos, el dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender dispositivos de comunicación inalámbrica, tales como equipos inalámbricos de mano, los denominados radiotéfonos celulares o por satélite, o dispositivos inalámbricos cualesquiera que puedan comunicar información de vídeo por un canal de comunicación 16, en cuyo caso el canal de comunicación 16 es inalámbrico. Las técnicas de esta divulgación, sin embargo, que conciernen a la codificación conjunta, p. ej., la señalización y / o interpretación conjunta de un valor conjuntamente señalado, y valores de diferencia de vectores de movimiento, que pueden tener precisión de sub-píxel, no están necesariamente limitadas a aplicaciones o configuraciones inalámbricas. Por ejemplo, estas técnicas pueden aplicarse a difusiones de televisión por el aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo por Internet, vídeo digital codificado que está codificado en un medio de almacenamiento, o a otros escenarios. En consecuencia, el canal de comunicación 16 puede comprender cualquier combinación de medios inalámbricos, cableados o de almacenamiento, adecuados para la transmisión o almacenamiento de datos de vídeo codificados.

En el ejemplo de la FIG. 1, el dispositivo de origen 12 incluye un origen de vídeo 18, un codificador de vídeo 20, un modulador / demodulador (módem) 22 y un transmisor 24. El dispositivo de destino 14 incluye un receptor 26, un módem 28, un descodificador de vídeo 30 y un dispositivo de visualización 32. De acuerdo a esta divulgación, el codificador de vídeo 20 del dispositivo de origen 12 puede ser configurado para aplicar las técnicas para la señalización de diferencias de vectores de movimiento, que pueden tener precisión de sub-píxel. En otros ejemplos, un dispositivo de origen y un dispositivo de destino pueden incluir otros componentes o disposiciones. Por ejemplo, el dispositivo de origen 12 puede recibir datos de vídeo desde un origen de vídeo 18 externo, tal como una cámara externa. Análogamente, el dispositivo de destino 14 puede mantener interfaces con un dispositivo de visualización externo, en lugar de incluir un dispositivo de visualización integrado.

30 El sistema ilustrado 10 de la FIG. 1 es tan solo un ejemplo. Las técnicas para la señalización conjunta de la precisión de diferencias de vectores de movimiento pueden ser realizadas por cualquier dispositivo de codificación y / o decodificación de vídeo digital. Aunque, en general, las técnicas de esta divulgación son realizadas por un dispositivo codificador de vídeo, las técnicas también pueden ser realizadas por un codificador / descodificador de vídeo, habitualmente mencionado como un "CODEC". Además, las técnicas de esta divulgación también pueden ser realizadas por un preprocesador de vídeo. El dispositivo de origen 12 y el dispositivo de destino 14 son tan solo ejemplos de tales dispositivos de codificación en los cuales el dispositivo de origen 12 genera datos de vídeo codificados para su transmisión al dispositivo de destino 14. En algunos ejemplos, los dispositivos 12, 14 pueden funcionar de una manera esencialmente simétrica, de modo que cada uno de los dispositivos 12, 14 incluya componentes de codificación y decodificación de vídeo. Por tanto, el sistema 10 puede dar soporte a la transmisión de vídeo unidireccional o bidireccional entre los dispositivos de vídeo 12, 14, p. ej., para la transmisión de flujos de vídeo, la reproducción de vídeo, la difusión de vídeo o la video-telefonía.

El origen de vídeo 18 del dispositivo de origen 12 puede incluir un dispositivo de captura de vídeo, tal como una cámara de vídeo, un archivo de vídeo que contiene vídeo previamente capturado y / o una fuente de vídeo desde un proveedor de contenidos de vídeo. Como una alternativa adicional, el origen de vídeo 18 puede generar datos basados en gráficos de ordenador como el vídeo de origen, o una combinación de vídeo en vivo, vídeo archivado y vídeo generado por ordenador. En algunos casos, si el origen de vídeo 18 es una cámara de vídeo, el dispositivo de origen 12 y el dispositivo de destino 14 pueden formar los denominados fono-cámaras o fono-videos. Como se ha mencionado anteriormente, sin embargo, las técnicas descritas en esta divulgación pueden ser aplicables a la codificación de vídeo en general, y pueden ser aplicadas a aplicaciones inalámbricas y / o cableadas. En cada caso, el vídeo capturado, pre-capturado o generado por ordenador puede ser codificado por un codificador de vídeo 20. La información de vídeo codificado puede luego ser modulada por el módem 22 de acuerdo a un estándar de comunicación, y transmitida al dispositivo de destino 14 mediante el transmisor 24. El módem 22 puede incluir diversos mezcladores, filtros, amplificadores u otros componentes diseñados para la modulación de señales. El transmisor 24 puede incluir circuitos diseñados para transmitir datos, incluyendo amplificadores, filtros y una o más antenas.

55 El receptor 26 del dispositivo de destino 14 recibe información por el canal 16, y el módem 28 desmodula la información. Nuevamente, el proceso de codificación de vídeo puede implementar una o más de las técnicas descritas en la presente memoria para señalar diferencias de vectores de vídeo, que pueden tener precisión de sub-píxel. La información comunicada por el canal 16 puede incluir información sintáctica definida por el codificador de vídeo 20, que también es

usada por el descodificador de vídeo 30, que incluye elementos sintácticos que describen características y / o procesamiento de macro-bloques y de otras unidades codificadas, p. ej., los GOP. El dispositivo de visualización 32 exhibe los datos de vídeo descodificados a un usuario, y puede comprender cualquiera entre una amplia variedad de dispositivos de visualización, tales como un tubo de rayos catódicos (CRT), un dispositivo de visualización de cristal líquido (LCD), un dispositivo de visualización de plasma, un dispositivo de visualización de diodos emisores de luz orgánica (OLED) u otro tipo de dispositivo de visualización.

En el ejemplo de la FIG. 1, el canal de comunicación 16 puede comprender cualquier medio de comunicación inalámbrico o cableado, tal como un espectro de frecuencia de radio (RF) o una o más líneas de transmisión físicas, o cualquier combinación de medios inalámbricos y cableados. El canal de comunicación 16 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 canal de comunicación 16 representa generalmente cualquier medio de comunicación adecuado, o una colección de distintos medios de comunicación, para transmitir datos de vídeo desde el dispositivo de origen 12 al dispositivo de destino 14, incluyendo cualquier combinación adecuada de medios cableados o inalámbricos. El canal de comunicación 16 puede incluir encaminadores, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo de origen 12 al dispositivo de destino 14. En otros ejemplos, el dispositivo de origen 12 puede almacenar datos codificados sobre un medio de almacenamiento, en lugar de transmitir los datos. Análogamente, el dispositivo de destino 14 puede ser configurado para extraer datos codificados desde un medio de almacenamiento.

El codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo a un estándar de compresión de vídeo, tal como el venidero estándar de Codificación de Vídeo de Alta Eficacia (HEVC) de ITU-T, también mencionado como "H.265". El HEVC no ha sido declarado como definitivo, por lo que el codificador de vídeo 20 y el descodificador 30 pueden funcionar de acuerdo al último borrador del HEVC, mencionado como el Modelo de Prueba de Codificación de Vídeo de Alta Eficacia (HM). Las técnicas de esta divulgación, sin embargo, no están limitadas a ningún estándar específico de codificación. Otros ejemplos incluyen las normas MPEG-2 e ITU-T H.263, y la ITU-T H.264. Aunque no se muestran en la FIG. 1, en algunos aspectos, cada uno entre el codificador de vídeo 20 y el descodificador de vídeo 30 puede estar integrado con un codificador y descodificador de audio, y puede incluir las unidades adecuadas de MUX-DEMUX, u otro hardware y software, para gestionar la codificación, tanto de audio como de vídeo, en un flujo común de datos o en flujos individuales de datos. Si corresponde, las unidades de MUX-DEMUX pueden ser conformes al protocolo multiplexador ITU H.223, o a otros protocolos tales como el protocolo de datagramas de usuario (UDP).

El estándar HEVC está siendo actualmente formulado por el Grupo de Expertos en Codificación de Vídeo (VCEG) de ITU-T, junto con el Grupo de Expertos en Películas (MPEG) de ISO / IEC, como el producto de una asociación colectiva conocida como el Equipo Colaborador Conjunto sobre Codificación de Vídeo (JCT-VC). El HM supone varias capacidades de dispositivos de codificación de vídeo sobre dispositivos de acuerdo a estándares previos de codificación, tales como el ITU-T H.264 / AVC. Por ejemplo, mientras que el H.264 proporciona nueve modalidades de codificación intra-predictiva, el HM proporciona hasta treinta y cuatro modalidades de codificación intra-predictivas.

Cada uno entre el codificador de vídeo 20 y el descodificador de vídeo 30 puede ser implementado como uno cualquiera entre una amplia 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 compuertas programables en el terreno (FPGA), lógica discreta, software, hardware, firmware o combinaciones cualesquiera de los mismos. Cada uno entre el codificador de vídeo 20 y el descodificador de vídeo 30 puede ser incluido en uno o más codificadores o descodificadores, cualquiera de los cuales puede estar integrado como parte de un codificador / descodificador (CODEC) combinado en una respectiva cámara, ordenador, dispositivo móvil, dispositivo de abonado, dispositivo difusor, equipo de sobremesa, servidor o similares.

Una secuencia de vídeo incluye habitualmente una serie de tramas de vídeo. Un grupo de imágenes (GOP) comprende generalmente una serie de una o más tramas de vídeo. Un GOP puede incluir datos sintácticos en una cabecera del GOP, en una cabecera de una o más tramas del GOP, o en otra parte, que describen un cierto número de tramas incluidas en el GOP. Cada trama puede incluir datos sintácticos de trama que describen una modalidad de codificación para la respectiva trama. El codificador de vídeo 20 opera habitualmente sobre bloques de vídeo, también mencionados como CU, dentro de tramas de vídeo individuales, a fin de codificar los datos de vídeo. Un bloque de vídeo puede corresponder a una LCU o a una partición de una LCU. Los bloques de vídeo pueden tener tamaños fijos o variables, y pueden diferir de tamaño de acuerdo a un estándar de codificación especificado. Cada trama de vídeo puede incluir una pluralidad de tajadas. Cada tajada puede incluir una pluralidad de LCU, que pueden estar dispuestas en particiones, también mencionadas como sub-CU.

Como ejemplo, el estándar ITU-T H.264 da soporte a la intra-predicción en diversos tamaños de bloque, tales como 16 por 16, 8 por 8, o 4 por 4 para componentes de luma, y 8x8 para componentes de croma, así como la inter-predicción en diversos tamaños de bloque, tales como 16x16, 16x8, 8x16, 8x8, 8x4, 4x8 y 4x4 para componentes de luma y los correspondientes tamaños ajustados a escala para los componentes de croma. En esta divulgación, "NxN" y "N por N" pueden ser usados de forma intercambiable para referirse a las dimensiones de píxeles del bloque, en términos de

dimensiones verticales y horizontales, p. ej., 16x16 píxeles o 16 por 16 píxeles. En general, un bloque de 16x16 tendrá 16 píxeles en una dirección vertical ($y = 16$) y 16 píxeles en una dirección horizontal ($x = 16$). Análogamente, un bloque de NxN tiene generalmente N píxeles en una dirección vertical y N píxeles en una dirección horizontal, donde N representa un valor entero no negativo. Los píxeles en un bloque pueden ser dispuestos en filas y columnas. Además, los bloques no necesariamente necesitan tener el mismo número de píxeles tanto en la dirección horizontal como en la dirección vertical. Por ejemplo, los bloques pueden comprender NxM píxeles, donde M no es necesariamente igual a N.

El HEVC se refiere a un bloque de datos de vídeo como una unidad de codificación (CU), que puede incluir una o más unidades de predicción (PU) y / o una o más unidades de transformación (TU). Los datos sintácticos dentro de un flujo de bits pueden definir una máxima unidad de codificación (LCU), que es una máxima unidad de codificación en términos del número de píxeles. En general, una CU tiene un propósito similar al de un macro-bloque de H.264, excepto porque una CU no tiene una distinción de tamaño. Por tanto, una CU puede ser dividida en sub-CU. En general, las referencias en esta divulgación a una CU pueden referirse a una máxima unidad de codificación de una imagen, o a una sub-CU de una LCU. Una LCU puede ser dividida en sub-CU, y cada sub-CU puede ser adicionalmente dividida en sub-CU. Los datos sintácticos para un flujo de bits pueden definir un máximo número de veces que puede ser dividida una LCU, mencionado como la profundidad de CU. En consecuencia, un flujo de bits también puede definir una mínima unidad de codificación (SCU). Esta divulgación también usa el término "bloque" para referirse a cualquiera entre una CU, una PU o una TU.

Una LCU puede estar asociada a una estructura de datos de cuadi-árbol. En general, una estructura de datos de cuadi-árbol incluye un nodo por CU, donde un nodo raíz corresponde a la LCU. Si una CU se divide en cuatro sub-CU, el nodo correspondiente a la CU incluye cuatro nodos hoja, cada uno de los cuales corresponde a una de las sub-CU. Cada nodo de la estructura de datos de cuadi-árbol puede proporcionar datos sintácticos para la CU correspondiente. Por ejemplo, un nodo en el cuadi-árbol puede incluir un indicador de división, que indica si la CU correspondiente al nodo está o no dividida en sub-CU. Los elementos sintácticos para una CU pueden ser definidos recursivamente, y pueden depender de si la CU está o no dividida en sub-CU. Si una CU no está adicionalmente dividida, se menciona como una CU hoja. En esta divulgación, 4 sub-CU de una CU hoja también serán mencionadas como CU hojas, aunque no haya ninguna división explícita de la CU hoja original. Por ejemplo, si una CU de tamaño 16x16 no se divide adicionalmente, las cuatro sub-CU de 8x8 también serán mencionadas como CU hojas, aunque la CU de 16x16 nunca fuera dividida.

Además, las TU de las CU hojas también pueden ser asociadas a las respectivas estructuras de datos de cuadi-árbol. Es decir, una CU hoja puede incluir un cuadi-árbol que indica cómo la CU hoja está dividida en las TU. Esta divulgación se refiere al cuadi-árbol que indica cómo está dividida una LCU como un cuadi-árbol de CU, y al cuadi-árbol que indica cómo está dividida una CU hoja en las TU, como un cuadi-árbol de TU. El nodo raíz de un cuadi-árbol de TU corresponde generalmente a una CU hoja, mientras que el nodo raíz de un cuadi-árbol de CU corresponde generalmente a una LCU. Las TU del cuadi-árbol de TU que no están divididas son mencionadas como TU hojas.

Una CU hoja puede incluir una o más unidades de predicción (PU). En general, una PU representa toda, o una parte de, la CU correspondiente, y puede incluir datos para extraer una muestra de referencia para la PU. Por ejemplo, cuando la PU está codificada en inter-modalidad, la PU puede incluir datos que definen un vector de movimiento para la PU. Los datos que definen el vector de movimiento pueden describir, por ejemplo, un componente horizontal del vector de movimiento, un componente vertical del vector de movimiento, una resolución para el vector de movimiento (p. ej., precisión de un cuarto de píxel o precisión de un octavo de píxel), una trama de referencia a la cual apunta el vector de movimiento y / o una lista de referencia (p. ej., lista 0 o lista 1) para el vector de movimiento. Los datos para la CU hoja que definen la(s) PU también pueden describir, por ejemplo, la división de la CU en una o más PU. Las modalidades de división pueden diferir, según que la CU esté no codificada, codificada en modalidad intra-predictiva, o codificada en modalidad inter-predictiva. Para la intra-codificación, una PU puede ser tratada igual que una unidad de transformación de hoja, descrita más adelante.

Una CU hoja puede incluir una o más unidades de transformación (TU). Las unidades de transformación pueden ser especificadas usando una estructura de cuadi-árbol de TU, según lo expuesto anteriormente. Es decir, un indicador de división puede indicar si una CU hoja está o no dividida en cuatro unidades de transformación. Luego, cada unidad de transformación puede ser adicionalmente dividida en 4 sub-TU. Cuando una TU no está adicionalmente dividida, puede ser mencionada como una TU hoja. En general, para la intra-codificación, todas las TU hojas pertenecientes a una CU hoja comparten la misma modalidad intra-predictiva. Es decir, la misma modalidad intra-predictiva se aplica generalmente para calcular los valores predichos para todas las TU de una CU hoja. Para la intra-codificación, un codificador de vídeo puede calcular un valor residual para cada TU hoja, usando la modalidad intra-predictiva, como una diferencia entre la parte de los valores predictivos correspondientes a la TU y el bloque original. El valor residual puede ser transformado, cuantizado y escaneado. Para la inter-codificación, un codificador de vídeo puede realizar la predicción al nivel de las PU y puede calcular un residuo para cada PU. Los valores residuales correspondientes a una CU hoja pueden ser transformados, cuantizados y escaneados. Para la inter-codificación, una TU hoja puede ser más grande o más pequeña que una PU. Para la intra-codificación, una PU puede ser co-situada con una correspondiente TU hoja. En algunos ejemplos, el tamaño máximo de una TU hoja puede ser el tamaño de la correspondiente CU hoja.

En general, esta divulgación usa los términos CU y TU para referirse a una CU hoja y a una TU hoja, respectivamente, a menos que se indique lo contrario. En general, las técnicas de esta divulgación se refieren a la transformación, cuantización, escaneo y codificación por entropía de los datos de una CU. Como ejemplo, las técnicas en esta divulgación incluyen la selección de una transformación, a usar para transformar un valor residual de un bloque intra-predicho, en base a la modalidad de intra-predicción usada para predecir el bloque. Esta divulgación también usa el término “transformación direccional” o “transformación diseñada” para referirse a una transformación de ese tipo que depende de la dirección de modalidad intra-predictiva. Es decir, un codificador de vídeo puede seleccionar una transformación direccional a aplicar a una unidad de transformación (TU). Como se ha observado anteriormente, la intra-predicción incluye la predicción de una TU de una CU actual de una imagen a partir de las CU y las TU previamente codificadas de la misma imagen. Más específicamente, un codificador de vídeo puede intra-predicir una TU actual de una imagen usando una modalidad intra-predictiva específica.

Durante las modalidades intra-predictivas, el codificador de vídeo 20 puede determinar una resolución, tal como una resolución de 1/4 (un cuarto) o 1/8 (un octavo) de píxel, para un vector de movimiento específico de una PU. Esta resolución puede ser determinada usando una técnica que intenta minimizar la magnitud del error entre un predictor de vectores de movimiento, y un vector de movimiento calculado para la PU de la CU, que puede ser un vector de movimiento usado para predecir un desplazamiento de una trama previamente codificada.

El codificador de vídeo 20 puede determinar una precisión para el predictor de vectores de movimiento, así como un desplazamiento del vector, que puede incluir un componente x, p_x, y un componente y, p_y. El codificador de vídeo 20 puede calcular una diferencia de vector de movimiento, correspondiente a una diferencia entre el predictor de vectores de movimiento y el vector de movimiento calculado. La diferencia de vector de movimiento (mencionada como una “MVD”) también puede tener un componente x, mvd_x, y un componente y, mvd_y. En base a la magnitud de mvd_x y de mvd_y, a la precisión de los vectores (es decir, precisión de un cuarto, o un octavo de píxel), y a un valor de umbral, el codificador de vídeo 20 puede ser configurado para codificar conjuntamente uno o más valores referidos a los componentes x e y del valor de diferencia de vector de movimiento.

De acuerdo a las técnicas de esta divulgación, el codificador de vídeo 20 y / o el decodificador de vídeo 30 pueden ser configurados para codificar conjuntamente información referida a diferencias de vectores de movimiento para un bloque. La codificación conjunta puede referirse a técnicas de codificación en las cuales las informaciones referidas a los componentes x e y de un valor de diferencia de vector de movimiento están intercaladas. La información puede incluir información indicativa de si una magnitud, p. ej., un valor absoluto, de los componentes es o no mayor que cero, un signo para los componentes cuando el valor absoluto del componente correspondiente es mayor que cero, e información indicativa del valor absoluto del componente cuando el valor absoluto es mayor que cero. La codificación conjunta de los uno o más valores de la diferencia de vector de movimiento puede reducir la complejidad de un flujo de bits de codificación de vídeo, lo que puede proporcionar las ventajas descritas en la presente memoria. De manera similar al codificador de vídeo 20, el decodificador de vídeo 30 puede recibir un flujo de bits de vídeo codificado y puede funcionar de una manera esencialmente recíproca a la del codificador de vídeo 20. Por ejemplo, el decodificador de vídeo 30 puede recibir valores codificados para una MVD, decodificar los valores codificados conjuntamente y calcular un vector de movimiento para un bloque, como una suma de la MVD y el predictor de vector de movimiento para el bloque.

En general, el componente x de un valor de diferencia de vector de movimiento puede ser mencionado como “MVD_x”, mientras que el componente y del valor de diferencia de vector de movimiento puede ser mencionado como “MVD_y”. El valor de diferencia de vector de movimiento para un bloque de datos de vídeo, p. ej., una PU, puede ser descrito por <MVD_x, MVD_y>. En general, MVD_x corresponde a la diferencia entre un componente x de un vector de movimiento para el bloque y un componente x de un predictor de vector de movimiento seleccionado para el bloque. Análogamente, MVD_y corresponde a la diferencia entre un componente y de un vector de movimiento para el bloque y un componente y de un predictor de vector de movimiento seleccionado para el bloque. De esta manera, el codificador de vídeo 20 puede calcular MVD_x calculando la diferencia entre el componente x del vector de movimiento para el bloque y el componente x del predictor de vector de movimiento para el bloque, y MVD_y calculando la diferencia entre el componente y del vector de movimiento para el bloque y el componente y del predictor de vector de movimiento para el bloque. De manera similar, el decodificador de vídeo 30 puede reconstruir el vector de movimiento para el bloque añadiendo MVD_x al componente x del predictor de vector de movimiento, y añadiendo MVD_y al componente y del predictor de vector de movimiento. En otras palabras, haciendo que MV_x y MV_y representen al componente x y al componente y del vector de movimiento, respectivamente:

$$MVD_x = MV_x - p_x; y \tag{1}$$

$$MVD_y = MV_y - p_y. \tag{2}$$

En algunos ejemplos, el codificador de vídeo 20 puede codificar, p. ej., señalar, información para el valor de diferencia de vector de movimiento para el bloque, de manera conjunta. Por ejemplo, el codificador de vídeo 20 puede codificar información indicativa de si un valor absoluto de MVD_x, es decir, |MVD_x|, es o no mayor que cero, seguida por

información indicativa de si un valor absoluto de MVD_y , es decir, $|MVD_y|$, es o no mayor que cero. Tal información puede incluir, por ejemplo, indicadores representativos de si los componentes x e y del valor de MVD son o no mayores que cero, respectivamente. El codificador de vídeo 20 también puede señalar un valor representativo de un signo para MVD_x , después de señalar el indicador que representa si MVD_y es o no mayor que cero, suponiendo que el indicador que representa si MVD_x es o no mayor que cero indica que MVD_x es, de hecho, mayor que cero. El codificador de vídeo 20 no necesita señalar un signo para MVD_x cuando $|MVD_x|$ es igual a cero.

A continuación del valor indicativo del signo para MVD_x (cuando se señala) o de la información que indica si $|MVD_y|$ es o no mayor que cero (cuando el signo de MVD_x no está señalado), el codificador de vídeo 20 puede señalar un signo para MVD_y , suponiendo nuevamente que $|MVD_y|$ es mayor que cero. Además, el codificador de vídeo 20 puede luego señalar información representativa de los valores absolutos de MVD_x y / o de MVD_y , según que cualquiera entre MVD_x y MVD_y , o ambos, tenga(n) valores absolutos mayores que cero. Nuevamente, la información representativa de los valores absolutos de MVD_x y MVD_y no necesariamente debe ser señalizada cuando los valores previamente señalizados indican que los valores absolutos de cualquiera entre MVD_x y MVD_y , o de ambos, son iguales a cero.

De manera similar, el descodificador de vídeo 30 puede codificar (p. ej., descodificar e interpretar) los datos representativos de si MVD_x tiene o no un valor absoluto mayor que cero, seguidos por los datos representativos de si MVD_y tiene o no un valor absoluto mayor que cero. Cuando los datos indican que el valor absoluto de MVD_x es mayor que cero, el descodificador de vídeo 30 puede ser configurado para analizar luego sintácticamente información representativa de un signo para MVD_x . Después de analizar sintácticamente la información representativa del signo para MVD_x , o después de determinar que $|MVD_x|$ es igual a cero, el descodificador de vídeo 30 puede ser configurado para analizar sintácticamente información representativa de un signo para MVD_y , cuando los datos indican que el valor absoluto de MVD_y es mayor que cero. De manera similar, el descodificador de vídeo 30 puede luego analizar sintácticamente información representativa de los valores absolutos para MVD_x y MVD_y , respectivamente, suponiendo nuevamente que los valores absolutos de MVD_x y MVD_y son mayores que cero.

El descodificador de vídeo 30 puede ser configurado para no esperar información relativa al signo y al valor absoluto de cualquier componente para el cual los datos indican que el componente tiene un valor absoluto no mayor que cero (es decir, igual a cero). Es decir, el descodificador de vídeo 30 puede ser configurado para analizar sintácticamente otros datos de vídeo, p. ej., otros datos para el bloque, sin recibir o codificar datos para los signos y valores absolutos de componentes de MVD que tienen valores absolutos iguales a cero, según lo indicado por la información anteriormente señalizada.

La codificación conjunta de componentes x e y de esta manera puede lograr ciertas ventajas sobre la codificación de datos para componentes x e y por separado. Por ejemplo, la codificación conjunta de los componentes x e y puede aumentar el caudal durante la codificación por entropía. Como un ejemplo particular, la codificación conjunta de valores de diferencia de vectores de movimiento puede aumentar el caudal de la codificación por entropía cuando el codificador de vídeo 20 o el descodificador 30 usa CABAC para realizar la codificación por entropía. Cuando la información representativa del valor absoluto de los componentes x e y y de valores de diferencia de vectores de movimiento se codifica conjuntamente, el codificador de vídeo 20, o el descodificador 30, puede ser capaz de codificar por entropía entre sí los componentes x e y del valor de diferencia de vector de movimiento, usando la modalidad de omisión de CABAC. La modalidad de omisión de CABAC puede mejorar el caudal de codificación por entropía. La codificación por entropía de la información codificada conjuntamente, representativa de los componentes x e y , puede permitir al codificador 20 o al descodificador 30 codificar por entropía ambos elementos sintácticos en sucesión usando la modalidad de omisión de CABAC, lo que puede mejorar las prestaciones de la codificación por entropía, con respecto a la codificación individual por entropía de los componentes x e y . Aunque se describe en el ejemplo precedente con respecto a CABAC, la codificación conjunta de los componentes x e y y de valores de diferencia de vectores de movimiento también puede aumentar la eficacia y / o el caudal de la codificación por entropía al usar técnicas de codificación por entropía, tales como VLC y CAVLC.

A continuación de la codificación intra-predictiva o inter-predictiva, para producir datos predictivos y datos residuales, y a continuación de transformaciones cualesquiera (tales como la transformación entera de 4×4 o de 8×8 , usada en la norma H.264 / AVC, o una transformación de coseno discreta, DCT) para producir coeficientes de transformación, puede realizarse la cuantización de los coeficientes de transformación. La cuantización se refiere generalmente a un proceso en el cual los coeficientes de transformación son cuantizados para reducir, posiblemente, la cantidad de datos usados para representar los coeficientes. El proceso de cuantización puede reducir la profundidad de bits asociada a algunos de, o todos, los coeficientes. Por ejemplo, un valor de n bits puede ser redondeado hacia abajo hasta un valor de m bits durante la cuantización, donde n es mayor que m .

A continuación de la cuantización, puede realizarse la codificación por entropía de los datos cuantizados, p. ej., de acuerdo a la codificación de longitud variable adaptativa en contenido (CAVLC), la codificación aritmética binaria adaptativa según el contexto (CABAC) u otra metodología de codificación por entropía. Una unidad de procesamiento configurada para la codificación por entropía, u otra unidad de procesamiento, puede realizar otras funciones de

procesamiento, tales como la codificación de longitud cero de rachas de coeficientes cuantizados y / o la generación de información sintáctica tal como los valores de patrones de bloques codificados (CBP), el tipo de macro-bloque, la modalidad de codificación, el tamaño máximo de macro-bloque para una unidad codificada (tal como una trama, una tajada, un macro-bloque o una secuencia) o similares.

5 El codificador de vídeo 20 puede además enviar datos sintácticos, tales como datos sintácticos basados en bloques, datos sintácticos basados en tramas y datos sintácticos basados en GOP, al descodificador de vídeo 30, p. ej., en una cabecera de trama, una cabecera de bloque, una cabecera de tajada o una cabecera de GOP. Los datos sintácticos de GOP pueden describir un cierto número de tramas en el GOP respectivo, y los datos sintácticos de tramas pueden indicar una modalidad de codificación / protección usada para codificar la trama correspondiente.

10 Cada uno entre el codificador de vídeo 20 y el descodificador de vídeo 30 puede ser implementado como cualquiera entre una amplia variedad de circuitos codificadores o descodificadores adecuados, según corresponda, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), formaciones de compuertas programables en el terreno (FPGA), circuitos lógicos discretos, software, hardware, firmware o combinaciones cualesquiera de los mismos. Cada uno entre el codificador de vídeo 20 y el descodificador de vídeo 30
15 puede ser incluido en uno o más codificadores o descodificadores, cada uno de los cuales puede estar integrado como parte de un codificador / descodificador (CODEC) de vídeo combinado. Un aparato que incluye el codificador de vídeo 20 y / o el descodificador de vídeo 30 puede comprender un circuito integrado, un microprocesador y / o un dispositivo de comunicación inalámbrica, tal como un teléfono celular.

20 La FIG. 2 es un diagrama de bloques que ilustra un ejemplo del codificador de vídeo 20 que puede implementar técnicas para señalar conjuntamente vectores de movimiento, que pueden tener precisión de sub-píxel. El codificador de vídeo 20 puede realizar la intra-codificación y la inter-codificación de bloques dentro de tramas de vídeo, incluyendo las CU, o las sub-CU de las CU. La intra-codificación se apoya en la predicción espacial para reducir o eliminar la redundancia espacial en el vídeo dentro de una trama de vídeo dada. La inter-codificación se apoya en la predicción temporal para reducir o eliminar la redundancia temporal en el vídeo dentro de tramas adyacentes de una secuencia de vídeo. La intra-modalidad (modalidad I) puede referirse a cualquiera entre varias modalidades de compresión de base espacial, y las inter-modalidades, tales como la predicción unidireccional (modalidad P) o la predicción bidireccional (modalidad B), pueden referirse a cualquiera entre varias modalidades de compresión de base temporal. Aunque los componentes para la codificación de inter-modalidad están ilustrados en la FIG. 2, debería entenderse que el codificador de vídeo 20 puede
25 además incluir componentes para la codificación de intra-modalidad. Sin embargo, tales componentes no están ilustrados, con fines de brevedad y claridad.

30 Según se muestra en la FIG. 2, el codificador de vídeo 20 recibe un bloque de vídeo actual dentro de una trama de vídeo que va a ser codificada. En el ejemplo de la FIG. 2, el codificador de vídeo 20 incluye la unidad de compensación de movimiento 44, la unidad de estimación de movimiento 42, la memoria de tramas de referencia 64, el sumador 50, la unidad de procesamiento de transformación 52, la unidad de cuantización 54 y la unidad de codificación por entropía 56. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 también incluye la unidad de cuantización inversa 58, la unidad de transformación inversa 60 y el sumador 62. Un filtro desbloqueador (no mostrado en la FIG. 2) también puede ser incluido para filtrar fronteras de bloques para eliminar distorsiones de bloqueo del vídeo reconstruido. Si se desea, el filtro desbloqueador filtrará habitualmente la salida del sumador 62.

35 Durante el proceso de codificación, el codificador de vídeo 20 recibe una trama, tajada o CU de vídeo para codificar. La trama o tajada puede ser dividida en múltiples bloques de vídeo. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 realizan la codificación inter-predictiva del bloque de vídeo recibido con respecto a uno o más bloques en una o más tramas de referencia, para proporcionar compresión temporal. La unidad de intra-predicción 46 puede realizar la codificación intra-predictiva del bloque de vídeo recibido con respecto a uno o más bloques vecinos en la misma trama o tajada que el bloque a codificar, para proporcionar compresión espacial. De acuerdo a las técnicas descritas más adelante, la unidad de estimación de movimiento 42 puede determinar una precisión para un predictor de vectores de movimiento, que puede ser determinada en base a una modalidad de intra-predicción o de inter-predicción que la unidad de selección de modalidad 40 ha determinado previamente. La unidad de estimación de movimiento 42, u otra unidad del codificador de vídeo 20, tal como la unidad de codificación por entropía 56, puede determinar además una diferencia de vector de movimiento correspondiente a una diferencia entre un predictor de vector
40 de movimiento y un vector de movimiento calculado para una PU de la trama, tajada o CU recibida. La unidad de codificación por entropía 56 puede además codificar información relacionada con la resolución, el signo y la amplitud de una diferencia de vector de movimiento, así como otra información relacionada con la codificación conjunta de un vector de movimiento, según se describe más adelante.

45 La unidad de selección de modalidad 40 puede seleccionar una de las modalidades de codificación, intra o inter, p. ej., en base a los resultados de errores, y proporcionar el bloque intra-codificado o inter-codificado resultante al sumador 50 para generar datos de bloques residuales, y al sumador 62 para reconstruir el bloque codificado, para su uso como una trama de referencia. Además, la unidad de selección de modalidad 40 puede seleccionar una precisión de vector de movimiento

para un vector de movimiento. Por ejemplo, la unidad de selección de modalidad 40 puede seleccionar una precisión de vector de movimiento para el vector de movimiento en base a la optimización de distorsión de tasa (RDO), asociada a vectores de movimiento de precisión de un cuarto de sub-píxel y de un octavo de sub-píxel. Los vectores de movimiento con precisión de un cuarto de sub-píxel pueden requerir menos bits para codificar, con respecto a vectores de movimiento con precisión de un octavo de píxel. Además, un bloque predictivo indicado con un vector de movimiento codificado con precisión de un cuarto de píxel puede dar como resultado una mayor cantidad de información residual, en comparación con un bloque predictivo codificado con precisión de un octavo de sub-píxel. El codificador de vídeo 20 puede realizar la RDO a fin de optimizar el número de bits (es decir, la tasa de bits), usada para codificar un vector de movimiento específico, con respecto a la información residual (distorsión) asociada a una diferencia entre el bloque predictivo indicado por el vector de movimiento y el bloque que está siendo codificado actualmente. En base al resultado de la RDO, la unidad de selección de modalidad 40 puede seleccionar la precisión del vector de movimiento, p. ej., precisión de un cuarto o de un octavo de sub-píxel, que optimiza el equilibrio entre distorsión y tasa.

La unidad de compensación de movimiento 44 puede ser necesaria para calcular sub-píxeles, tales como sub-píxeles de tramas de referencia, en distintas precisiones, p. ej., de sub-píxeles de un octavo y de un cuarto. A fin de interpolar sub-píxeles, la unidad de compensación de movimiento 44 puede utilizar una amplia variedad de técnicas. Como ejemplos, la unidad de compensación de movimiento 44 puede utilizar interpolación bilineal o utilizar filtros de respuesta finita (FIR) de N tomas para interpolar un sub-píxel. Cuando un dispositivo, tal como una unidad de compensación de movimiento 44, calcula un valor para un píxel fraccionario promediando dos píxeles o sub-píxeles, puede redondear y / o ajustar a escala el valor resultante. En algunos casos, la unidad de compensación de movimiento 44 puede promediar valores para dos sub-píxeles que son el resultado de promediar hasta un píxel sub-entero.

La unidad de compensación de movimiento 44 puede calcular valores para más posiciones de píxeles sub-enteros, tales como posiciones de píxeles de un octavo, aplicando filtros de interpolación a conjuntos de soporte. El soporte se refiere generalmente a valores para uno o más píxeles de referencia, p. ej., píxeles en una línea o región común. Los píxeles pueden corresponder a posiciones de píxel completas o a posiciones de píxeles sub-enteros que fueron calculadas previamente. En algunos ejemplos, la unidad de compensación de movimiento 44 puede calcular valores para píxeles sub-enteros usando interpolación bilineal, y puede usar filtros similares de interpolación bilineal para calcular valores para dos o más posiciones distintas de píxeles sub-enteros, aplicando los uno o más filtros de interpolación bilineal a distintos conjuntos de soporte para las respectivas posiciones de píxeles sub-enteros.

En algunos otros casos, la unidad de compensación de movimiento 44 puede utilizar un filtro de respuesta finita (FIR) de N tomas para interpolar valores de sub-píxel. Un FIR, tal como un filtro de Wiener de 6 tomas o de 12 tomas, puede utilizar los valores de píxeles de soporte vecinos para interpolar un valor de píxel sub-entero. Un píxel de soporte es un valor de píxel o sub-píxel usado como una entrada para el FIR. Un FIR puede tener una o más dimensiones. En un FIR unidimensional, un dispositivo tal como la unidad de compensación de movimiento 44 puede aplicar un filtro a un cierto número de píxeles o sub-píxeles de soporte, en una línea, por ejemplo, horizontalmente, verticalmente, o en cierto ángulo. A diferencia de un FIR unidimensional, que puede usar píxeles de soporte en una línea recta, un FIR bidimensional puede usar píxeles o sub-píxeles de soporte vecinos que formen un cuadrado o rectángulo para calcular el valor de píxel interpolado.

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 de movimiento es el proceso de generar vectores de movimiento, que estiman el movimiento para bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de un bloque predictivo dentro de una trama de referencia predictiva (u otra unidad codificada) con respecto al bloque actual que está siendo codificado dentro de la trama actual (u otra unidad codificada). Un bloque predictivo es un bloque del que se halla que coincide estrechamente con el bloque a codificar, en términos de diferencia de píxeles, que puede ser determinada por la suma de la diferencia absoluta (SAD), la suma de la diferencia al cuadrado (SSD) u otras métricas de diferencia. Un vector de movimiento también puede indicar el desplazamiento de una partición de un macro-bloque. La compensación de movimiento puede implicar capturar o generar el bloque predictivo en base al vector de movimiento determinado por la estimación de movimiento. Nuevamente, la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden estar funcionalmente integradas, en algunos ejemplos.

La unidad de estimación de movimiento 42 calcula un vector de movimiento para el bloque de vídeo de una trama inter-codificada, comparando el bloque de vídeo con bloques de vídeo de una trama de referencia en la memoria de tramas de referencia 64. La unidad de compensación de movimiento 44 puede también interpolar píxeles sub-enteros de la trama de referencia, p. ej., una trama I o una trama P. Como ejemplo, los vectores de movimiento pueden ser predichos a partir de dos listas de tramas de referencia: la lista 0, que incluye tramas de referencia que tienen un orden de visualización anterior al de una trama actual que está siendo codificada, y la lista 1, que incluye tramas de referencia que tienen un orden de visualización posterior al de la trama actual que está siendo codificada. Por lo tanto, los datos almacenados en la memoria de tramas de referencia 64 pueden ser actualizados de acuerdo a estas listas.

La unidad de estimación de movimiento 42 compara bloques de una o más tramas de referencia, procedentes de la

memoria de tramas de referencia 64, con un bloque que va a ser codificado de una trama actual, p. ej., una trama P o una trama B. Cuando las tramas de referencia en la memoria de tramas de referencia 64 incluyen valores para píxeles sub-enteros, un vector de movimiento calculado por la unidad de estimación de movimiento 42 puede referirse a una ubicación de píxel sub-entero en una de las tramas de referencia almacenadas en la memoria de tramas de referencia 64. La unidad de estimación de movimiento 42 y / o la unidad de compensación de movimiento 44 también pueden ser configuradas para calcular valores para posiciones de píxel sub-entero de tramas de referencia almacenadas en la memoria de tramas de referencia 64 si no se almacena ningún valor para posiciones de píxel sub-entero en la memoria de tramas 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. El bloque de trama de referencia identificado por un vector de movimiento puede ser mencionado como un bloque predictivo. El vector de movimiento que indica el desplazamiento del bloque predictivo con respecto al bloque actual que está siendo codificado en la trama actual puede tener un componente x de mvd_x y un componente y de mvd_y .

La unidad de compensación de movimiento 44 puede calcular datos de predicción en base al bloque predictivo. El codificador de vídeo 20 forma un bloque de vídeo residual restando los datos de predicción procedentes de la unidad de compensación de movimiento 44 al bloque de vídeo original que está siendo codificado. El sumador 50 representa el componente, o los componentes, que realiza(n) esta operación de resta. La unidad de procesamiento de transformación 52 aplica una transformación, tal como una transformación de coseno discreta (DCT) o una transformación conceptualmente similar, al bloque residual, produciendo un bloque de vídeo que comprende valores de coeficientes de transformación residual. La unidad de procesamiento de transformación 52 puede realizar otras transformaciones, tales como las definidas por el estándar H.264, que son conceptualmente similares a la DCT. Las transformaciones de ondículas, las transformaciones enteras, las transformaciones de sub-banda u otros tipos de transformación, también podrían usarse. En cualquier caso, la unidad de procesamiento de transformación 52 aplica la transformación al bloque residual, produciendo un bloque de coeficientes de transformación residual. La transformación puede convertir la información residual, de un dominio de valores de píxel a un dominio de transformación, tal como un dominio de frecuencia. La unidad de cuantización 54 cuantiza los coeficientes de transformación residual para reducir adicionalmente la tasa de bits. El proceso de cuantización puede reducir la profundidad de bits asociada a algunos de, o todos, los coeficientes. El grado de cuantización puede ser modificado ajustando un parámetro de cuantización.

A continuación de la cuantización, la unidad de codificación por entropía 56 codifica por entropía los coeficientes de transformación cuantizados. Por ejemplo, la unidad de codificación por entropía 56 puede realizar la codificación de longitud variable adaptativa en contenido (CAVLC), la codificación aritmética binaria adaptativa según el contexto (CABAC), u otra técnica de codificación por entropía. La codificación conjunta de los componentes x e y de los valores de diferencia de vectores de movimiento puede permitir a la unidad de codificación por entropía 56 lograr un mayor caudal de codificación por entropía al usar la CABAC. La unidad de codificación por entropía 56 puede usar la modalidad de omisión de la CABAC para aumentar el caudal de la codificación por entropía de los componentes x e y conjuntamente codificados de los valores de diferencia de vectores de movimiento, con respecto a la codificación por entropía de los componentes x e y, codificados por separado, de valores de diferencia de vectores de movimiento. A continuación de la codificación por entropía por parte de la unidad de codificación por entropía 56, el vídeo codificado puede ser transmitido a otro dispositivo o archivado para su posterior transmisión o extracción. En el caso de la codificación aritmética binaria adaptativa según el contexto, el contexto puede estar basado en macro-bloques vecinos.

En algunos casos, la unidad de codificación por entropía 56, u otra unidad del codificador de vídeo 20, puede ser configurada para realizar otras funciones de codificación, además de la codificación por entropía. Por ejemplo, la unidad de codificación por entropía 56 puede ser configurada para determinar los valores de CBP para los macro-bloques y particiones. Además, en algunos casos, la unidad de codificación por entropía 56 puede realizar la codificación de longitud de rachas de los coeficientes en un macro-bloque o una partición del mismo. En particular, la unidad de codificación por entropía 56 puede aplicar un escaneo en zigzag u otro patrón de escaneo para escanear los coeficientes de transformación en un macro-bloque o partición, y codificar rachas de ceros para una compresión adicional. La unidad de codificación por entropía 56 también puede construir información de cabecera con los elementos sintácticos adecuados, para su transmisión en el flujo de bits de vídeo codificado.

De acuerdo a las técnicas de esta divulgación, la unidad de cuantización 54 y la unidad de codificación por entropía 56 pueden codificar conjuntamente vectores de movimiento, que pueden tener precisión de sub-píxel, según se describe más adelante. En algunos ejemplos, la unidad de cuantización 45 y la unidad de codificación por entropía 56 pueden señalar uno o más valores indicadores que representan si los componentes x y / o los componentes y de un valor de diferencia de vector de movimiento son cero o no, uno o más valores que representan el signo de los componentes x y / o los componentes y cuando los componentes tienen valores absolutos no iguales a cero, e información que representa el valor absoluto de la magnitud de los componentes x e y del valor de diferencia de vector de movimiento cuando los componentes tienen valores absolutos no iguales a cero. La unidad de cuantización 45 y la unidad de codificación por entropía 56 también pueden señalar un indicador de resolución de movimiento, que puede representar la resolución de sub-píxeles del vector de movimiento, así como un valor de umbral relacionado tanto con la magnitud como con la

precisión del vector de movimiento.

La unidad de cuantización inversa 58 y la unidad de transformación inversa 60 aplican, respectivamente, la cuantización inversa y la transformación inversa para reconstruir el bloque residual en el dominio del píxel, p. ej., para su uso posterior como un bloque de referencia. La unidad de compensación de movimiento 44 puede calcular un bloque de referencia añadiendo el bloque residual a un bloque predictivo de una de las tramas de la memoria de tramas de referencia 64. La unidad de compensación de movimiento 44 también puede aplicar uno o más filtros de interpolación al bloque residual reconstruido, para calcular valores de píxeles sub-enteros para su uso en la estimación de movimiento. El sumador 62 añade el bloque residual reconstruido al bloque de predicción compensado en movimiento, producido por la unidad de compensación de movimiento 44, para producir un bloque de vídeo reconstruido para su almacenamiento en la memoria de tramas de referencia 64. El bloque de vídeo reconstruido puede ser usado por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 como un bloque de referencia para inter-codificar un bloque en una posterior trama de vídeo.

El codificador de vídeo 20 representa un ejemplo de un codificador de vídeo configurado para codificar datos de vídeo representativos de si un valor absoluto de un componente x de un valor de diferencia de vector de movimiento, para un bloque actual de datos de vídeo, es o no mayor que cero, y para codificar datos de vídeo representativos de si un valor absoluto de un componente y del valor de diferencia de vector de movimiento es o no mayor que cero. El codificador de vídeo 20 puede ser configurado para codificar datos de vídeo representativos del valor absoluto del componente x del valor de diferencia de vector de movimiento cuando el valor absoluto del componente x del valor de diferencia de vector de movimiento es mayor que cero, y para codificar datos de vídeo representativos del valor absoluto del componente y del valor de diferencia de vector de movimiento cuando el valor absoluto del componente y del valor de diferencia de vector de movimiento es mayor que cero. El codificador de vídeo 20 también puede ser configurado para codificar datos de vídeo representativos de un signo del componente x del valor de diferencia de vector de movimiento cuando el valor absoluto del componente x del valor de diferencia de vector de movimiento es mayor que cero, y para codificar un signo del componente y del valor de diferencia de vector de movimiento cuando el valor absoluto del componente y del valor de diferencia de vector de movimiento es mayor que cero.

La FIG. 3 es un diagrama de bloques que ilustra un ejemplo del descodificador de vídeo 30, que descodifica una secuencia de vídeo codificado. En el ejemplo de la FIG. 3, el descodificador de vídeo 30 incluye una unidad de descodificación por entropía 70, una unidad de compensación de movimiento 72, una unidad de intra-predicción 74, una unidad de cuantización inversa 76, una unidad de transformación inversa 78, una memoria de tramas de referencia 82 y un sumador 80. El descodificador de vídeo 30, en algunos ejemplos, puede realizar un pase de descodificación, generalmente recíproco al pase de codificación descrito con respecto al codificador de vídeo 20 (FIG. 2). La unidad de compensación de movimiento 72 puede generar datos de predicción en base a vectores de movimiento recibidos desde la unidad de descodificación por entropía 70.

La unidad de compensación de movimiento 72 puede usar vectores de movimiento recibidos en el flujo de bits, incluyendo vectores de movimiento señalizados de acuerdo a las técnicas descritas en la presente memoria, para identificar un bloque de predicción en una de las tramas de referencia almacenadas en la memoria de tramas de referencia 82. La unidad de intra-predicción 74 puede usar modalidades de intra-predicción recibidas en el flujo de bits para formar un bloque de predicción a partir de bloques espacialmente adyacentes. La unidad de cuantización inversa 76 cuantiza inversamente, es decir, descuantiza, los coeficientes de bloques cuantizados proporcionados en el flujo de bits y descodificados por la unidad de descodificación por entropía 70. El proceso de cuantización inversa puede incluir un proceso convencional, por ejemplo, según lo definido por el estándar de descodificación H.264. El proceso de cuantización inverso también puede incluir el uso de un parámetro de cuantización QP_y , calculado por el codificador 50 para cada macro-bloque, para determinar un grado de cuantización y, análogamente, un grado de cuantización inversa que debería ser aplicado.

La unidad de descodificación por entropía 70 y la unidad de cuantización inversa 76 también pueden reconstruir los valores de diferencia de vectores de movimiento conjuntamente codificados, interpretando diversos elementos sintácticos que incluyen indicadores y otros valores en el flujo de bits codificado. Como ejemplo, la unidad de descodificación por entropía 70 y la unidad de cuantización inversa 76 pueden recibir uno o más valores que representan si los componentes x y / o los componentes y de una diferencia de vector de movimiento son o no mayores que cero, un indicador de resolución de movimiento, que puede indicar la resolución en sub-píxeles del vector de movimiento, y un valor de umbral, que puede indicar una magnitud y / o resolución máxima del valor de diferencia de vector de movimiento. En un ejemplo, la unidad de descodificación por entropía 70 puede descodificar los componentes x e y conjuntamente codificados del valor de diferencia de vector de movimiento, usando la CABAC en la modalidad de omisión. El uso de la modalidad de omisión de la CABAC puede permitir a la unidad de codificación por entropía 70 descodificar los valores de diferencia de vectores de movimiento conjuntamente codificados, con un caudal aumentado con respecto a la descodificación por entropía de los componentes x e y de los valores de diferencia de vectores de movimiento por separado. La unidad de descodificación por entropía 70 y la unidad de cuantización inversa 76 también pueden recibir uno o más valores que representan el signo de los componentes x y / o de los componentes y del valor de diferencia de vector de movimiento, y

valores que representan el valor absoluto de la magnitud de cada componente del valor de diferencia de vector de movimiento. La unidad de descodificación por entropía 70 y la unidad de cuantización inversa 76 pueden usar los valores que representan si los componentes x e y de los valores de diferencia de vectores de movimiento son o no mayores que cero, así como los valores que representan el valor absoluto de la magnitud de cada componente del valor de diferencia de vector de movimiento, para reconstruir la amplitud, signo, precisión, componentes x e y de los valores de diferencia de vectores de movimiento.

La unidad de transformación inversa 58 aplica una transformación inversa, p. ej., una DCT inversa, una transformación entera inversa o un proceso de transformación inversa conceptualmente similar, a los coeficientes de transformación, a fin de producir bloques residuales en el dominio de píxeles. La unidad de compensación de movimiento 72 produce bloques compensados en movimiento, realizando posiblemente la interpolación en base a filtros de interpolación. Los identificadores para filtros de interpolación a usar para la estimación de movimiento con precisión de sub-píxel pueden ser incluidos en los elementos sintácticos. La unidad de compensación de movimiento 72 puede usar filtros de interpolación como los usados por el codificador de vídeo 20 durante la codificación del bloque de vídeo, para calcular valores interpolados para píxeles sub-enteros de un bloque de referencia.

La unidad de compensación de movimiento 72 puede determinar los filtros de interpolación usados por el codificador de vídeo 20 de acuerdo a la información sintáctica recibida, y usar los filtros de interpolación para producir bloques predictivos.

La unidad de compensación de movimiento 72 usa algo de la información sintáctica para determinar tamaños de macro-bloques usados para codificar la(s) trama(s) de la secuencia de vídeo codificado, información de partición que describe cómo se divide cada macro-bloque de una trama de la secuencia de vídeo codificado, modalidades que indican cómo se codifica cada partición, una o más tramas de referencia (y listas de tramas de referencia) para cada macro-bloque o partición inter-codificados, y otra información, para descodificar la secuencia de vídeo codificada. Como ejemplo, la unidad de compensación de movimiento 72 puede recibir un valor de MVD y un predictor de vector de movimiento. La unidad de compensación de movimiento 72 puede además determinar un vector calculado para un bloque actual como una suma del valor de MVD y el predictor de vector de movimiento del bloque actual. El vector de movimiento calculado para el bloque actual puede indicar además la ubicación de un bloque de predicción. En algunos ejemplos, la unidad de compensación de movimiento 72 puede calcular valores de píxeles para el bloque de predicción, p. ej., cuando el vector de movimiento tiene precisión de sub-píxel, tal como precisión de un cuarto o un octavo de píxel.

El sumador 80 suma los bloques residuales con los correspondientes bloques de predicción generados por la unidad de compensación de movimiento 72 o la unidad de intra-predicción, para formar bloques descodificados. Si se desea, también puede aplicarse un filtro desbloqueador para filtrar los bloques descodificados, a fin de eliminar distorsiones de bloqueo. Los bloques de vídeo descodificado son luego almacenados en la memoria de tramas de referencia 82, que proporciona bloques de referencia para la posterior compensación de movimiento y también produce vídeo descodificado para su presentación en un dispositivo de visualización (tal como el dispositivo de visualización 32 de la FIG. 1).

En base al vector de movimiento para el bloque actual, la unidad de compensación de movimiento puede determinar un bloque predictivo a partir de una o más tramas almacenadas en la memoria de tramas de referencia 82. La unidad de compensación de movimiento 72 puede ser necesaria para calcular sub-píxeles del bloque predictivo que genera la unidad de compensación de movimiento 72. Los bloques de predicción tales como los sub-píxeles de tramas de referencia, en diversas precisiones, p. ej., sub-píxeles de un octavo y un cuarto. A fin de interpolar sub-píxeles, la unidad de compensación de movimiento 72 puede utilizar una amplia variedad de técnicas. Como ejemplos, la unidad de compensación de movimiento 72 puede utilizar la interpolación bilineal o utilizar filtros de respuesta finita (FIR) de N tomas para interpolar un sub-píxel. Cuando un dispositivo tal como la unidad de compensación de movimiento 72 calcula un valor para un píxel fraccionario promediando dos píxeles o sub-píxeles, puede redondear y / o ajustar a escala el valor resultante. En algunos casos, la unidad de compensación de movimiento 72 puede promediar valores para dos sub-píxeles, que son el resultado de promediar hasta un píxel sub-entero.

La unidad de compensación de movimiento 72 puede calcular valores para posiciones de píxel sub-entero, tales como posiciones de un octavo de píxel, aplicando filtros de interpolación a conjuntos de soporte. El soporte se refiere generalmente a valores para uno o más píxeles de referencia, p. ej., píxeles en una línea o región común. Los píxeles pueden corresponder a posiciones de píxel completo o a posiciones de píxel sub-entero que fueron calculadas previamente. En algunos ejemplos, la unidad de compensación de movimiento 72 puede calcular valores para píxeles sub-enteros usando interpolación bilineal, y puede usar filtros similares de interpolación bilineal para calcular valores para dos o más posiciones distintas de píxel sub-entero, aplicando los uno o más filtros de interpolación bilineal a distintos conjuntos de soporte para las respectivas posiciones de píxel sub-entero.

En algunos otros casos, la unidad de compensación de movimiento 42 puede utilizar un filtro de respuesta finita (FIR) de N tomas para interpolar valores de sub-píxeles. Un FIR, tal como un filtro de Wiener de 6 tomas o de 12 tomas, puede utilizar valores de píxeles de soporte vecinos para interpolar un valor de píxel sub-entero. Un píxel de soporte es un píxel o

valor de sub-píxel usado como una entrada para el FIR. Un FIR puede tener una o más dimensiones. En un FIR unidimensional, un dispositivo tal como la unidad de compensación de movimiento 72 puede aplicar un filtro a un cierto número de píxeles de soporte, o sub-píxeles, en una línea, por ejemplo, horizontalmente, verticalmente, o en un cierto ángulo. A diferencia de un FIR unidimensional, que puede usar píxeles de soporte en una línea recta, un FIR bidimensional puede usar píxeles o sub-píxeles de soporte vecinos que forman un cuadrado o rectángulo para calcular el valor de píxel interpolado.

El descodificador de vídeo 30 representa un ejemplo de un codificador de vídeo configurado para descodificar datos de vídeo representativos de si un valor absoluto de un componente x de un valor de diferencia de vector de movimiento para un bloque actual de datos de vídeo es o no mayor que cero, y para descodificar datos de vídeo representativos de si un valor absoluto de un componente y del valor de diferencia de vector de movimiento es o no mayor que cero. El descodificador de vídeo 30 puede ser configurado para descodificar datos de vídeo representativos del valor absoluto del componente x del valor de diferencia de vector de movimiento cuando el valor absoluto del componente x del valor de diferencia de vector de movimiento es mayor que cero, y para descodificar datos de vídeo representativos del valor absoluto del componente y del valor de diferencia de vector de movimiento cuando el valor absoluto del componente y del valor de diferencia de vector de movimiento es mayor que cero. El descodificador de vídeo 30 también puede ser configurado para descodificar datos de vídeo representativos de un signo del componente x del valor de diferencia de vector de movimiento cuando el valor absoluto del componente x del valor de diferencia de vector de movimiento es mayor que cero, y para descodificar un signo del componente y del valor de diferencia de vector de movimiento cuando el valor absoluto del componente y del valor de diferencia de vector de movimiento es mayor que cero.

La FIG. 4 es un diagrama de flujo que ilustra un procedimiento ejemplar para codificar conjuntamente los componentes x e y de un valor de diferencia de vector de movimiento y codificar por entropía un bloque actual. En el procedimiento 150, la unidad de estimación de movimiento 42 del codificador de vídeo 20 puede predecir un bloque actual (152). Como parte de la predicción del bloque actual, la unidad de estimación de movimiento 42 puede calcular un vector de movimiento. El vector de movimiento puede indicar un desplazamiento de un bloque predictivo con respecto al bloque actual. El desplazamiento del vector de movimiento puede tener componentes x e y, MV_x y MV_y . El bloque predictivo puede coincidir estrechamente con el bloque actual en términos de diferencia de píxeles, que puede ser determinada por la suma de diferencias absolutas (SAD), la suma de diferencias al cuadrado (SSD) u otras métricas de diferencia. En el caso en que la trama actual está inter-predicha, el bloque predictivo puede comprender un bloque de una trama distinta, tal como una entre una o más tramas de referencia en la memoria de tramas de referencia 64 de la FIG. 2.

La unidad de compensación de movimiento 44, u otra unidad del codificador de vídeo 20, puede entonces determinar un predictor de vector de movimiento para el vector de movimiento del bloque actual (154). Un predictor de vector de movimiento es un vector de movimiento que la unidad de compensación de movimiento 44 usa para calcular valores de diferencia de vectores de movimiento para el bloque actual. En algunos ejemplos, el predictor de vector de movimiento puede ser el vector de movimiento de uno entre una pluralidad de bloques que son vecinos de la PU del bloque actual. La unidad de compensación de movimiento 44 también puede determinar el predictor de vector de movimiento a partir de un bloque co-situado de una de las tramas de referencia almacenadas en la memoria de tramas de referencia 64. El predictor de vector de movimiento puede tener un componente x de p_x y un componente y de p_y . El codificador de vídeo 20 puede determinar el predictor de vector de movimiento de modo que el predictor de vector de movimiento minimice la diferencia del vector de movimiento para el bloque actual y del predictor de vector de movimiento.

En base al predictor de vector de movimiento, la unidad de compensación de movimiento 44 del codificador de vídeo 20, u otra unidad del codificador de vídeo 20, tal como la unidad de codificación por entropía 56, puede calcular un valor de diferencia de vector de movimiento para el bloque actual (156). La unidad de compensación de movimiento 44 puede calcular la diferencia de vector de movimiento para el bloque actual como una diferencia del vector de movimiento para el bloque actual y el predictor de vector de movimiento para el bloque actual. De tal modo, la unidad de compensación de movimiento 44 puede calcular el predictor de vector de movimiento para el bloque actual como $\langle p_x - MV_x, p_y - MV_y \rangle$, donde p_x y p_y son los componentes x e y del predictor de vector de movimiento, y MV_x y MV_y son los componentes x e y del vector de movimiento para el bloque actual.

La unidad de codificación por entropía 56 puede luego codificar conjuntamente elementos del valor de diferencia de vector de movimiento para el bloque actual (158). Por ejemplo, la unidad de codificación por entropía 56 puede codificar conjuntamente elementos del valor de diferencia de vector de movimiento, que incluyen los componentes x e y de la diferencia de vector de movimiento, según lo explicado con respecto a las FIGs. 6 y 7 en mayor detalle más adelante. Los elementos de la diferencia de vector de movimiento también pueden incluir uno o más valores representativos de si los componentes x e y de la diferencia de vector de movimiento son o no cero, un valor representativo del valor del signo de los componentes x y / o de los componentes y del valor de diferencia de vector de movimiento, y un valor representativo del valor absoluto del componente del valor de diferencia de vector de movimiento, como algunos ejemplos no limitadores.

La unidad de procesamiento de transformación 52, o la unidad de compensación de movimiento 44, puede calcular un

- 5 bloque residual para el bloque actual (160). El bloque residual puede comprender diferencias de píxeles entre el bloque de vídeo predictivo, que está indicado por el vector de movimiento para el bloque actual, y el bloque de vídeo actual. La unidad de procesamiento de transformación 52 puede luego aplicar una transformación, tal como una transformación de coseno discreta (DCT), al bloque residual para el bloque actual, mientras que la unidad de cuantización 54 puede cuantizar los coeficientes de transformación procedentes de la unidad de procesamiento de transformación 52 (162).
- 10 La unidad de codificación por entropía 56 puede escanear los coeficientes transformados, resultantes de la transformación de 162. En algunos casos, la unidad de codificación por entropía 56 puede aplicar un escaneo en zigzag u otro patrón de escaneo para escanear los coeficientes de transformación del bloque residual transformado (164). La unidad de codificación por entropía 56 puede codificar por entropía los coeficientes que comprenden rachas de ceros, resultantes del escaneo de 164, por ejemplo, usando la codificación por longitud de rachas, para una compresión adicional (166). La unidad de codificación por entropía 56 puede luego emitir los coeficientes codificados por entropía y el valor de MVD codificado conjuntamente (168). Es decir la unidad de codificación por entropía 56 puede emitir datos conjuntamente codificados para componentes x e y del valor de MVD, además de los coeficientes de transformación cuantizados y codificados por entropía.
- 15 Las técnicas de la FIG. 4 pueden ser generalmente realizadas por cualquier unidad de procesamiento o procesador, ya sea implementados en hardware, software, firmware, o una combinación de los mismos y, cuando son implementados en software o firmware, el hardware correspondiente puede ser proporcionado para ejecutar instrucciones para el software o el firmware. Con fines de ejemplo, las técnicas de la FIG. 4 se describen con respecto a un dispositivo de codificación de vídeo, que puede incluir componentes esencialmente similares a los del codificador de vídeo 20 (FIGs. 1 y 2), aunque debería entenderse que otros dispositivos pueden ser configurados para realizar técnicas similares. Además, las etapas 20 ilustradas en la FIG. 4 pueden ser realizadas en un orden distinto, o en paralelo, y pueden añadirse etapas adicionales y omitirse ciertas etapas, sin apartarse de las técnicas de esta divulgación.
- 25 La FIG. 5 es un diagrama de flujo que ilustra un procedimiento de señalización por separado por un vector de movimiento de un valor de diferencia de vector de movimiento, de acuerdo a las técnicas descritas en el modelo de prueba del HEVC (HM3.0, también mencionado como HM3). De acuerdo al procedimiento propuesto en HM3, e ilustrado en la FIG. 5, los valores de mvd_x y mvd_y , correspondientes a los componentes x e y de un valor de diferencia de vector de movimiento, son señalizados por separado. El procedimiento 180 ilustra cómo uno o más elementos sintácticos, relacionados con mvd_x , pueden ser señalizados, y el procedimiento 192 ilustra cómo uno o más elementos sintácticos, relacionados con mvd_y , pueden ser señalizados.
- 30 En el procedimiento 180, una unidad del codificador de vídeo 20, tal como la unidad de compensación de movimiento 44, o la unidad de codificación por entropía 56, puede señalar un indicador de $mvd_x == 0$ (182). El indicador de $mvd_x == 0$ representa si la diferencia entre los componentes x de un vector de movimiento calculado para una PU de una CU, y un predictor de vector de movimiento para el vector de movimiento, son o no iguales y, en consecuencia, si la correspondiente diferencia entre los dos componentes x es o no igual a cero.
- 35 Si el indicador de $mvd_x == 0$ está activado, entonces mvd_x , el componente x de la MVD, es igual a cero (rama "Sí" de 184), y el procedimiento termina (190). Sin embargo, si mvd_x no es cero (rama "No" de 184), el codificador de vídeo 20 puede señalar abs_mvd_x-1 (186). Abs_mvd_x-1 puede representar el valor absoluto del valor de mvd_x , menos uno. El codificador de vídeo 20 puede además señalar un valor que representa el signo de mvd_x (188). El signo puede ser, por ejemplo, positivo o negativo.
- 40 De manera similar, en el procedimiento 192, un codificador de vídeo, tal como el codificador de vídeo 20 de la FIG. 1, puede señalar un indicador de $mvd_y == 0$ (194). El indicador de $mvd_y == 0$ representa si la diferencia entre los componentes y de un vector de movimiento calculado para una PU de una CU, y un predictor de vector de movimiento para el vector de movimiento, son o no iguales y, en consecuencia, si la correspondiente diferencia entre los dos componentes y es o no igual a cero.
- 45 Si el indicador de $mvd_y == 0$ está activado, entonces mvd_y , el componente y de la MVD, es igual a cero (rama "Sí" de 196) y el procedimiento termina (202). Sin embargo, si mvd_y no es cero (rama "No" de 196), el codificador de vídeo 20 puede señalar abs_mvd_y-1 (198). Abs_mvd_y-1 puede ser un valor que representa el valor absoluto del valor de mvd_y , menos uno. El codificador de vídeo 20 puede además señalar un valor que representa un signo de mvd_y (200). El signo puede ser, por ejemplo, positivo o negativo.
- 50 La FIG. 6 es un diagrama de flujo que ilustra un procedimiento ejemplar 220 de codificación conjunta de valores de diferencia de vectores de movimiento para codificar conjuntamente los componentes x e y de un valor de diferencia de vector de movimiento. En general, el procedimiento 220 de la FIG. 6 corresponde a la etapa 158 de la FIG. 4. De tal modo, el procedimiento 220 de la FIG. 6 representa un procedimiento ejemplar para codificar conjuntamente los datos de componentes x e y de un valor de diferencia de vector de movimiento. Además, se supone que antes de la realización del procedimiento ejemplar de la FIG. 6, la unidad de compensación de movimiento 44 del codificador de vídeo 20, u otra
- 55

unidad del codificador de vídeo 20, tal como la unidad de codificación por entropía 50 del codificador de vídeo 20, puede calcular un vector de movimiento para el bloque actual de datos de vídeo y calcular el valor de diferencia de vector de movimiento como una diferencia entre el vector de movimiento para el bloque actual de datos de vídeo y un predictor de vector de movimiento, determinado para el bloque actual de datos de vídeo, p. ej., según lo expuesto con respecto a la etapa 156 de la FIG. 4. Para codificar conjuntamente datos para los componentes x e y del valor de diferencia de vector de movimiento, la unidad de codificación por entropía 56 puede codificar información representativa de si el valor absoluto de mvd_x , el componente x del valor de diferencia de vector de movimiento, es o no mayor que cero (222). La unidad de codificación por entropía 56 puede luego codificar información representativa de si un valor absoluto de un componente y (mvd_y) del valor de diferencia de vector de movimiento es o no mayor que cero (224).

Cuando el valor absoluto del componente x del valor de diferencia de vector de movimiento es mayor que cero, la unidad de codificación por entropía 56 puede codificar información representativa del valor absoluto del componente x del valor de diferencia de vector de movimiento (226). Y cuando el valor absoluto del componente y del valor de diferencia de vector de movimiento es mayor que cero, la unidad de codificación por entropía 56 puede codificar información representativa del valor absoluto del componente y del valor de diferencia de vector de movimiento (228). En algunos ejemplos, la unidad de codificación por entropía 56 puede codificar la información representativa del valor absoluto de los componentes x e y de los valores de diferencia de vector de movimiento como valores absolutos de $mvd_x/2 - 1$, y $mvd_y/2 - 1$, respectivamente.

Cuando el valor absoluto del componente x del valor de diferencia de vector de movimiento es mayor que cero, la unidad de codificación por entropía 56 puede codificar un signo del componente x del valor de diferencia de vector de movimiento (230). Cuando el valor absoluto del componente y del valor de diferencia de vector de movimiento es mayor que cero, la unidad de codificación por entropía 56 puede codificar un signo del componente y del valor de diferencia de vector de movimiento (232).

Las técnicas de la FIG. 6 pueden ser generalmente realizadas por cualquier unidad de procesamiento o procesador, ya sea implementados en hardware, software, firmware o una combinación de los mismos y, cuando son implementados en software o firmware, el correspondiente hardware puede ser proporcionado para ejecutar instrucciones para el software o firmware. Con fines de ejemplo, las técnicas de la FIG. 6 se describen con respecto a un dispositivo de codificación de vídeo, que puede incluir componentes esencialmente similares a las del codificador de vídeo 20 (FIG. 1 y 2), aunque debería entenderse que otros dispositivos pueden ser configurados para realizar técnicas similares. Además, las etapas ilustradas en la FIG. 6 pueden ser realizadas en un orden distinto, o en paralelo, y pueden añadirse etapas adicionales y omitirse ciertas etapas, sin apartarse de las técnicas de esta divulgación.

De esta manera, el procedimiento de la FIG. 6 representa un ejemplo de un procedimiento que incluye codificar información representativa de si un valor absoluto de un componente x de un valor de diferencia de vector de movimiento para un bloque actual de datos de vídeo es o no mayor que cero, codificar información representativa de si un valor absoluto de un componente y del valor de diferencia de vector de movimiento es o no mayor que cero, cuando el valor absoluto del componente x del valor de diferencia de vector de movimiento es mayor que cero, codificar información representativa del valor absoluto del componente x del valor de diferencia de vector de movimiento, cuando el valor absoluto del componente y del valor de diferencia de vector de movimiento es mayor que cero, codificar información representativa del valor absoluto del componente y del valor de diferencia de vector de movimiento, cuando el valor absoluto del componente x del valor de diferencia de vector de movimiento es mayor que cero, codificar un signo del componente x del valor de diferencia de vector de movimiento y, cuando el valor absoluto del componente y del valor de diferencia de vector de movimiento es mayor que cero, codificar un signo del componente y del valor de diferencia de vector de movimiento. De tal modo, el procedimiento de la FIG. 6 representa un ejemplo de codificación conjunta de elementos de componentes x e y de un valor de diferencia de vector de movimiento, calculado para un vector de movimiento de un bloque actual de datos de vídeo.

La FIG. 7 es un diagrama de flujo que ilustra un procedimiento para señalar valores para valores de mvd_x y mvd_y . A diferencia de los procedimientos 180 y 192 de la FIG. 5, que ilustran la codificación de los componentes x e y de una MVD por separado, y de manera similar al procedimiento de la FIG. 6, el procedimiento de la FIG. 7 puede también permitir a un codificador codificar conjuntamente los componentes x e y de un valor de diferencia de vector de movimiento, es decir, mvd_x y mvd_y . En este ejemplo, un elemento sintáctico señalado en el flujo de bits por un codificador, tal como el codificador de vídeo 20 de las FIGs. 1 y 2, puede contener la magnitud, el signo y / u otra información relacionada con los componentes x e y de un valor de diferencia de vector de movimiento.

En el ejemplo de la FIG. 7, según se explica más adelante, el codificador de vídeo 20 puede codificar un indicador de resolución de movimiento, que es un valor representativo de si un vector de movimiento del bloque actual de datos de vídeo tiene una primera precisión de sub-píxel, p. ej., precisión de cuarto de píxel, o una segunda precisión de sub-píxel, p. ej., precisión de octavo de píxel. En este ejemplo, el indicador de resolución de movimiento tiene un valor de uno para indicar que el vector de movimiento del bloque actual tiene precisión de cuarto de píxel, y tiene un valor de cero para indicar que el indicador de resolución de movimiento tiene precisión de octavo de píxel. El codificador de vídeo 20 también

puede codificar un valor, "abs_mvd_igual_2", que representa si la magnitud de uno de los componentes, o de ambos, de la diferencia de vector de movimiento, tiene o no un valor absoluto de dos. El codificador de vídeo 20 puede asignar un valor a abs_mvd_igual_2 en base al ejemplo de la Tabla 1 dada a continuación.

5 En el ejemplo del procedimiento 240 de la FIG. 7, la unidad de compensación de movimiento 44, u otra unidad del codificador de vídeo 20, puede señalar un indicador de $mvd_x == 0$ (242), y un indicador de $mvd_y == 0$ (244). Estos indicadores pueden ser fijados cuando los correspondientes valores de mvd_x o mvd_y son iguales a cero. Si tanto mvd_y como mvd_x son cero (rama "Sí" de 246), no es necesario señalar ningún dato adicional, y el procedimiento 240 termina (266).

10 Si mvd_x o mvd_y no son cero (rama "No" de 246), la unidad de compensación de movimiento 44 puede señalar o codificar un indicador de resolución de movimiento (246). El indicador de resolución de movimiento (indicador_mvres) es un ejemplo de un elemento sintáctico que puede indicar una resolución de vector de movimiento (también mencionada como precisión de vector de movimiento) para un vector de movimiento correspondiente a los valores de diferencia de vectores de movimiento. El indicador de resolución de movimiento puede indicar si el vector de movimiento del bloque actual de datos de vídeo tiene una primera precisión de sub-píxel o una segunda precisión de sub-píxel, p. ej., precisión de un cuarto de píxel o precisión de un octavo de píxel.

20 El codificador de vídeo 20 también puede determinar un umbral para una trama actual, que también está asociado al vector de movimiento para el bloque actual, y que puede ser señalado en la cabecera de tajada o en otra estructura de datos, p. ej., un conjunto de parámetros de tajada (SPS) o un conjunto de parámetros de imagen (PPS). El codificador de vídeo 20 puede usar el umbral para restringir la amplitud de valores de diferencia de vectores de movimiento de octavo de píxel, y puede indicar que el vector de movimiento tiene la primera precisión de sub-píxel cuando tanto el componente x del valor de diferencia de vector de movimiento (mvd_x) como el componente y del valor de diferencia de vector de movimiento (mvd_y) son mayores que el umbral. Por ejemplo, según se muestra más adelante con respecto a la Tabla 1 dada a continuación, el codificador de vídeo (p. ej., el codificador de vídeo 20) puede restringir la magnitud de los componentes x e y de un valor de diferencia de vector de movimiento de precisión de octavo de píxel, a valores de uno o dos. El codificador de vídeo puede señalar el valor del umbral usando un indicador de umbral.

25 Si la MVD calculada para un componente x o y de un vector de movimiento para una PU es mayor que el umbral determinado, el indicador de resolución de movimiento puede ser fijado en el valor verdadero. Si el indicador de resolución de movimiento está fijado en el valor verdadero, el indicador que representa que el vector de movimiento tiene precisión de un cuarto de píxel. Si el valor de MVD es menor o igual que el umbral seleccionado, el codificador de vídeo puede fijar un valor para el indicador de resolución de movimiento, para indicar si un correspondiente vector de movimiento tiene precisión de un cuarto o de un octavo de píxel.

35 El procedimiento 240 puede comprender adicionalmente que el codificador de vídeo 20 determine si el indicador de resolución de movimiento es igual o no a uno (250). El indicador de resolución de movimiento puede ser igual a uno, por ejemplo, si la precisión del vector de movimiento es precisión de un cuarto. Si el indicador de resolución de movimiento se fija en uno, el codificador de vídeo 20 puede señalar o codificar un valor representativo del valor $abs_mvd_x/2 - 1$ si mvd_x no es cero, donde abs_mvd_x es un valor que representa el valor absoluto del componente x de la MVD (252). El codificador de vídeo 20 puede codificar de manera similar un valor representativo de $abs_mvd_y/2 - 1$ si mvd_y no es cero, donde abs_mvd_y es el valor absoluto del componente y de la MVD (254). En algunos ejemplos, el codificador de vídeo 20 puede señalar $abs_mvd_y/2 - 1$ si mvd_y no es cero, en lugar de si mvd_x no es cero.

40 Si el indicador de resolución de movimiento no es igual a uno (rama "No" de 250), el codificador de vídeo 20 puede determinar si el umbral es o no igual a uno. El umbral puede tener un valor de uno o dos en este ejemplo. En un ejemplo, el umbral puede tener un valor de dos. Debido a que el vector de movimiento solamente puede tener precisión de octavo de píxel cuando los valores de mvd_x y mvd_y no superan el umbral, los valores de mvd_x y de mvd_y están restringidos a uno o dos en este ejemplo. Además, si el umbral es igual a uno (rama "Sí" de 256), y el vector de movimiento tiene precisión de cuarto de píxel (según lo indicado por el indicador de resolución de movimiento, que en este ejemplo debe tener el valor verdadero a continuación de la etapa 250), el codificador de vídeo no necesita señalar ninguna información adicional para los valores de mvd. Es decir, puede determinarse que los valores de mvd_x y de mvd_y son cero o uno usando otra información. En particular, puede indicarse si mvd_x y mvd_y son cero o uno en base a si los valores del signo son señalizados o no en las etapas 248 y 264. Si el umbral es igual a dos (rama "No" de 256), el codificador de vídeo 20 puede determinar si alguno de los valores de mvd_x o de mvd_y es o no igual a cero. Si mvd_x o mvd_y es igual a cero (rama "Sí" de 260), el codificador de vídeo 20 puede proceder a señalar los signos del valor no nulo de mvd_x y mvd_y (258, 264). En algunos ejemplos, el codificador de vídeo 20 puede señalar adicionalmente un valor para el valor no nulo de mvd_x y mvd_y , y, por tanto, puede avanzar a la etapa 252, en lugar de a la 258.

55 Sin embargo, si tanto mvd_x como mvd_y son no nulos, es decir, ni mvd_x ni mvd_y tiene un valor de cero (rama "No" de 260), entonces el codificador de vídeo 20 puede señalar un indicador de abs_mvd_igual_a_2. El indicador de abs_mvd_igual_a_2 es un ejemplo de un valor conjuntamente codificado para mvd_x y mvd_y . Un codificador de vídeo

puede asignar un valor a `abs_mvd_igual_a_2` en base a la Tabla 1 dada a continuación.

TABLA 1

<code>abs_mvd_igual_a_2</code>	<code>abs_mvd_x</code>	<code>abs_mvd_y</code>
0	1	1
1	1	2
2	2	1

5 En base a los valores de `abs_mvd_x` y de `abs_mvd_y`, el codificador de vídeo 20 puede señalar el valor de `abs_mvd_igual_a_2` (262), p. ej., de acuerdo a la Tabla 1. El codificador de vídeo 20 puede señalar el valor de `abs_mvd_igual_a_2` usando binarización unaria. El codificador de vídeo 20 puede además señalar el signo de `mvd_x` si `mvd_x` no es cero (258) y, de manera similar, señalar el signo de `mvd_y` si `mvd_x` no es cero (264). En algunos ejemplos, el codificador de vídeo 20 puede señalar el signo de `mvd_y` si `mvd_y` no es cero.

10 En algunos ejemplos, el indicador de resolución de movimiento, `mvd_x == 0`, y el indicador de `mvd_y == 0` son codificados conjuntamente. Optativamente, y como ejemplo, pueden usarse códigos de VLC de acuerdo a la Tabla 2 dada a continuación:

TABLA 2

<code>mvd_x == 0</code>	<code>mvd_y == 0</code>	Indicador de resolución de movimiento	palabra de código de VLC
verdadero	verdadero	x	00
verdadero	falso	1	101
verdadero	falso	0	1110
falso	verdadero	1	100
falso	verdadero	0	110
falso	falso	0	1111
falso	falso	1	01

15 El codificador de vídeo 20 puede señalar el indicador de resolución de movimiento para indicar una entre una resolución de vector de movimiento de un cuarto y de un octavo de píxel. Por ejemplo, el codificador de vídeo 20 puede señalar la resolución de la diferencia de vector de movimiento entre 1/4 de píxel y 1/8 de píxel. El codificador de vídeo 20 puede señalar el indicador de resolución de movimiento para indicar una diferencia entre una precisión de vector de movimiento de un cuarto de píxel y de un octavo de píxel cuando la MVD es más pequeña que un valor de umbral. El codificador de vídeo 20 puede también señalar el indicador de resolución de movimiento para indicar una diferencia entre una precisión de vector de movimiento de un cuarto y un octavo de píxel cuando uno de los componentes, o ambos, de la MVD es mayor que un valor de umbral.

25 Un dispositivo de descodificación de vídeo, tal como el descodificador de vídeo 30, puede funcionar de una manera esencialmente recíproca al codificador de vídeo 20 para realizar las técnicas ilustradas en la FIG. 7. De esta manera, el descodificador de vídeo 30 puede seleccionar adaptativamente la precisión del vector de movimiento para cada vector de movimiento, en base a la recepción y descodificación de un valor que representa el vector de movimiento, tal como el indicador de resolución de movimiento. El descodificador de vídeo 30 también puede recibir y descodificar el valor de umbral a partir de (256). El descodificador de vídeo 30 puede recibir y descodificar el valor de umbral para cada trama de referencia, para restringir la amplitud de los valores de diferencia de vectores de movimiento de un octavo de píxel para la trama. El umbral puede ser recibido en la cabecera de tajada. El umbral puede tener valores de cero, uno o dos. El umbral no está restringido, y puede ser extendido a otros valores. Codificando información acerca de los componentes x e y de valores de diferencia de vector de movimiento, la complejidad de un flujo de bits de vídeo codificado puede ser reducida, lo que puede dar como resultado una menor tasa global de bits para los datos de vídeo codificado.

En un ejemplo, un dispositivo de codificación de vídeo, tal como el codificador de vídeo 20, o el decodificador de vídeo 30, puede codificar un valor, tal como el indicador de resolución de movimiento de la FIG. 7, que es representativo de si un vector de movimiento del bloque actual de datos de vídeo tiene una primera precisión de sub-píxel o una segunda precisión de sub-píxel, tal como una precisión de sub-píxel de un cuarto o de un octavo. La segunda precisión de sub-píxel puede ser mayor que la primera precisión de sub-píxel. Cuando el vector de movimiento tiene la segunda precisión de sub-píxel, p. ej., precisión de un octavo de píxel, cuando el componente x del valor de diferencia de vector de movimiento no es igual a cero, y cuando el componente y del valor de diferencia de vector de movimiento no es igual a cero, el codificador de vídeo 20, o el decodificador de vídeo 30, puede codificar información representativa del valor absoluto del componente x del valor de diferencia de vector de movimiento, y codificar información representativa del valor absoluto del componente y del valor de diferencia de vector de movimiento. El codificador de vídeo 20, o el decodificador de vídeo 30, puede codificar un valor conjuntamente codificado, representativo tanto del componente x del valor de diferencia de vector de movimiento como del componente y del valor de diferencia de vector de movimiento.

Las técnicas de la FIG. 7 pueden ser generalmente realizadas por cualquier unidad de procesamiento o procesador, ya sea implementados en hardware, software, firmware o una combinación de los mismos, y cuando son implementados en software o firmware, el hardware correspondiente puede ser proporcionado para ejecutar instrucciones para el software o el firmware. Con fines de ejemplificación, las técnicas de la FIG. 7 se describen con respecto a un dispositivo de codificación de vídeo, que puede incluir componentes esencialmente similares a las del codificador de vídeo 20 (FIGs. 1 y 2). El decodificador de vídeo 30 de las FIGs. 1 y 3 puede realizar operaciones generalmente recíprocas de las ilustradas y descritas en la FIG. 7 para decodificar un flujo de bits codificado de acuerdo a las técnicas ilustradas en la FIG. 7. También debería entenderse que otros dispositivos pueden ser configurados para realizar técnicas similares. Además, las etapas ilustradas en la FIG. 7 pueden ser realizadas en un orden distinto, o en paralelo, y pueden añadirse etapas adicionales y omitirse ciertas etapas, sin apartarse de las técnicas de esta divulgación.

El procedimiento de la FIG. 7 representa un ejemplo de un procedimiento que incluye determinar si se usa un vector de movimiento que tiene una primera precisión de sub-píxel o una segunda precisión de sub-píxel para codificar una unidad de predicción de una unidad de codificación de datos de vídeo, en donde la segunda precisión de sub-píxel es mayor que la primera precisión de sub-píxel; calcular un componente x y un componente y del vector de movimiento que tiene la primera precisión de sub-píxel o la segunda precisión de sub-píxel, en base a la determinación, cuando el componente x y el componente y tienen la segunda precisión de sub-píxel; determinar valores de diferencia de vector de movimiento en base a los valores del componente x y el componente y del vector de movimiento, y a los valores de un componente x predicho y de un componente y predicho para el vector de movimiento; y codificar un valor conjuntamente codificado, representativo de los valores de diferencia de vector de movimiento para el componente x y el componente y para el vector de movimiento.

Un procedimiento recíproco puede ser realizado por un decodificador de vídeo. El procedimiento recíproco, por ejemplo, puede incluir determinar si un vector de movimiento de una unidad de predicción de una unidad de codificación de datos de vídeo tiene una primera precisión de sub-píxel o una segunda precisión de sub-píxel, usando un valor de un indicador de resolución de movimiento para el vector de movimiento, en donde la segunda precisión de sub-píxel es mayor que la primera precisión de sub-píxel, cuando el vector de movimiento tiene la segunda precisión de sub-píxel; decodificar un valor conjuntamente codificado, representativo de valores de diferencia de vectores de movimiento para un componente x y un componente y del vector de movimiento; y decodificar el vector de movimiento en base a los valores de diferencia de vectores de movimiento y a los valores de un componente x predicho y un componente y predicho para el vector de movimiento, y formar datos de predicción para la unidad de codificación usando el vector de movimiento decodificado.

El codificador de vídeo 20, o el decodificador de vídeo 30, puede determinar la resolución del vector de movimiento para el bloque actual, en base al valor de umbral ilustrado en la FIG. 7. Más específicamente, el codificador de vídeo 20, o el decodificador de vídeo 30, puede determinar si el vector de movimiento para el bloque actual tiene la primera precisión de sub-píxel o la segunda precisión de sub-píxel, determinando un valor de umbral asociado al vector de movimiento. El codificador de vídeo 20, o el decodificador de vídeo 30, puede determinar que el vector de movimiento tiene la primera precisión de sub-píxel cuando tanto el componente x de un valor de diferencia de vector de movimiento del bloque corriente, como el componente y del valor de diferencia de vector de movimiento, son mayores que el umbral.

Cuando al menos uno entre el componente x del valor de diferencia de vector de movimiento y el componente y del valor de diferencia de vector de movimiento es menor o igual que el umbral, el codificador de vídeo 20, o el decodificador de vídeo 30, puede determinar si el vector de movimiento tiene la primera precisión de sub-píxel o la segunda precisión de sub-píxel, en base al valor del indicador de resolución de vector de movimiento.

Ha de reconocerse que, según el ejemplo, ciertos actos o sucesos de cualquiera de las técnicas descritas en la presente memoria pueden ser realizados en una secuencia distinta, pueden ser añadidos, combinados o descartados completamente (p. ej., no todos los actos o sucesos descritos son necesario para la práctica de las técnicas). Además, en ciertos ejemplos, los actos o sucesos pueden ser realizados simultáneamente, p. ej., mediante procesamiento de múltiples hebras, procesamiento de interrupciones o procesadores múltiples, en lugar de secuencialmente.

La FIG. 8 es un diagrama de flujo que ilustra un procedimiento ejemplar para descodificar conjuntamente los componentes x e y de un valor de diferencia de vector de movimiento y descodificar datos codificados por entropía para producir un bloque actual. Las etapas del procedimiento 280 de la FIG. 8 son generalmente recíprocas para el procedimiento 150 de la FIG. 4. En el procedimiento 280, la unidad de compensación de movimiento 72, u otra unidad del descodificador de vídeo 30, puede predecir un bloque actual (282). Como parte de la predicción del bloque actual, la unidad de compensación 72 puede descodificar componentes conjuntamente codificados de un valor de diferencia de vector de movimiento (284). El descodificador de vídeo puede descodificar los componentes conjuntamente codificados de la diferencia de vector de movimiento, de acuerdo a las técnicas del procedimiento 320 ilustrado en la FIG. 9. El valor de diferencia de vector de movimiento para el bloque actual puede tener un componente x de mvd_x y un componente y de mvd_y . Los componentes conjuntamente codificados del valor de diferencia de vector de movimiento pueden incluir componentes x e y del valor de diferencia de vector de movimiento para el bloque actual, así como otra información referida al valor de diferencia de vector de movimiento, tal como uno o más valores representativos de si los componentes x e y de la diferencia de vector de movimiento son o no cero, un valor representativo del valor del signo de los componentes x y / o de los componentes y del valor de diferencia de vector de movimiento, y un valor representativo del valor absoluto del componente del valor de diferencia de vector de movimiento, como algunos ejemplos no limitadores.

La unidad de compensación de movimiento 44 del codificador de vídeo 20 también puede recibir un predictor de vector de movimiento para el vector de movimiento del bloque actual (286). Un predictor de vector de movimiento es un vector de movimiento que la unidad de compensación de movimiento 72 usa para calcular valores de diferencia de vectores de movimiento para el bloque actual. En algunos ejemplos, el predictor de vector de movimiento puede ser el vector de movimiento de uno entre una pluralidad de bloques vecinos de la PU del bloque actual. La unidad de compensación de movimiento 72 también puede determinar el predictor de vector de movimiento a partir de un bloque co-situado de una de las tramas de referencia almacenadas en la memoria de tramas de referencia 82. El predictor de vector de movimiento puede tener un componente x de p_x y un componente y de p_y . El predictor de vector de movimiento puede haber sido determinado por el codificador de vídeo 20 durante la fase de codificación, de modo que el predictor de vector de movimiento minimice la diferencia del vector de movimiento para el bloque actual y el predictor de vector de movimiento. En base al predictor de vector de movimiento, la unidad de compensación de movimiento 72 del codificador de vídeo 20 puede calcular un vector de movimiento para el bloque actual (286). El descodificador de vídeo 30 puede calcular el vector de movimiento como una suma de los componentes de la diferencia de vector de movimiento y el predictor de vector de movimiento, $\langle p_x + mvd_x, p_y + mvd_y \rangle$ (288).

La unidad de descodificación por entropía 70 puede recibir datos codificados por entropía para el bloque actual (290), y descodificar datos, tales como los coeficientes transformados, codificados por longitud de rachas, de un bloque para reproducir los coeficientes transformados del bloque (292). La unidad de descodificación por entropía 70 puede escanear inversamente los coeficientes reproducidos para re-disponer los coeficientes procedentes de uno o más vectores unidimensionales de coeficientes en un bloque bidimensional de coeficientes transformados (294).

La unidad de cuantización inversa 76 puede cuantizar inversamente el bloque bidimensional de coeficientes transformados, y la unidad de transformación inversa 78 puede transformar inversamente los coeficientes inversamente cuantizados para producir un bloque de píxeles no transformados (296). El bloque residual puede comprender diferencias de píxeles entre el bloque de vídeo predictivo, que está indicado por el vector de movimiento para el bloque actual, y el bloque de vídeo actual. El descodificador de vídeo 30 puede luego combinar el bloque residual con el bloque predictivo para formar el bloque actual de la trama actual (298).

Las técnicas de la FIG. 8 pueden ser generalmente realizadas por cualquier unidad de procesamiento o procesador, ya sea implementados en hardware, software, firmware o una combinación de los mismos y, cuando son implementados en software o firmware, el hardware correspondiente puede ser proporcionado para ejecutar instrucciones para el software o firmware. Con fines de ejemplo, las técnicas de la FIG. 8 se describen con respecto a un dispositivo de descodificación de vídeo, que puede incluir componentes esencialmente similares a las del descodificador de vídeo 30 (FIGs. 1 y 3), aunque debería entenderse que otros dispositivos pueden ser configurados para realizar técnicas similares. Además, las etapas ilustradas en la FIG. 8 pueden ser realizadas en un orden distinto, o en paralelo, y pueden añadirse etapas adicionales y omitirse ciertas etapas, sin apartarse de las técnicas de esta divulgación.

La FIG. 9 es un diagrama de flujo que ilustra un procedimiento ejemplar para descodificar conjuntamente los componentes x e y de un valor de diferencia de vector de movimiento. Las etapas del procedimiento 320 de la FIG. 9 son generalmente recíprocas para las etapas del procedimiento 220 de la FIG. 6. Análogamente, el procedimiento 320 puede corresponder, en general, a la etapa 284 de la FIG. 8. En el procedimiento 320 de la FIG. 9, la unidad de compensación de movimiento 72 del descodificador de vídeo 30, u otra unidad del descodificador de vídeo 30, puede recibir una representación conjuntamente codificada de un vector de movimiento para el bloque actual de datos de vídeo. El descodificador de vídeo 30 también puede determinar un predictor de vector de movimiento, que un descodificador de vídeo, tal como el codificador de vídeo 20, puede haber determinado y codificado previamente. El descodificador de vídeo 30 puede calcular un vector de movimiento para el bloque actual como una suma o diferencia del predictor de vector de movimiento para el bloque actual de datos de vídeo y del predictor de vector de movimiento determinado para el bloque

actual de datos de vídeo.

Para determinar los componentes del valor de diferencia de vector de movimiento, la unidad de compensación de movimiento 72 del descodificador de vídeo 30 puede descodificar información representativa de si el valor absoluto de mvd_x , el componente x del valor de diferencia de vector de movimiento, es o no mayor que cero (322). La unidad de compensación de movimiento 72 puede entonces descodificar información representativa de si un valor absoluto de un componente y (mvd_y) del valor de diferencia de vector de movimiento es o no mayor que cero (324).

Cuando el valor absoluto del componente x del valor de diferencia de vector de movimiento es mayor que cero, la unidad de compensación de movimiento 72 puede descodificar información representativa del valor absoluto del componente x del valor de diferencia de vector de movimiento (326). Y cuando el valor absoluto del componente y del valor de diferencia de vector de movimiento es mayor que cero, la unidad de compensación de movimiento 72 puede descodificar información representativa del valor absoluto del componente y del valor de diferencia de vector de movimiento (328). En algunos ejemplos, el descodificador de vídeo 30 puede descodificar la información representativa del valor absoluto de los componentes x e y de los valores de diferencia de vectores de movimiento como los valores absolutos de $mvd_x/2 -1$ y $mvd_y/2 -1$, respectivamente.

Cuando el valor absoluto del componente x del valor de diferencia de vector de movimiento es mayor que cero, la unidad de compensación de movimiento 72 puede descodificar un signo del componente x del valor de diferencia de vector de movimiento (330). Cuando el valor absoluto del componente y del valor de diferencia de vector de movimiento es mayor que cero, la unidad de compensación de movimiento 72 puede descodificar un signo del componente y del valor de diferencia de vector de movimiento (332).

Las técnicas de la FIG. 9 pueden ser generalmente realizadas por cualquier unidad de procesamiento o procesador, ya sea implementados en hardware, software, firmware o una combinación de los mismos y, cuando son implementados en software o firmware, el hardware correspondiente puede ser proporcionado para ejecutar instrucciones para el software o firmware. Con fines de ejemplificación, las técnicas de la FIG. 9 se describen con respecto a un dispositivo de codificación de vídeo, que puede incluir componentes esencialmente similares a las del descodificador de vídeo 30 (FIGs. 1 y 3), aunque debería entenderse que otros dispositivos pueden ser configurados para realizar técnicas similares. Además, las etapas ilustradas en la FIG. 9 pueden ser realizadas en un orden distinto, o en paralelo, y pueden añadirse etapas adicionales y omitirse ciertas etapas, sin apartarse de las técnicas de esta divulgación.

De esta manera, el procedimiento de la FIG. 9 representa un ejemplo de un procedimiento que incluye codificar información representativa de si un valor absoluto de un componente x de un valor de diferencia de vector de movimiento para un bloque actual de datos de vídeo es o no mayor que cero, codificar información representativa de si un valor absoluto de un componente y del valor de diferencia de vector de movimiento es o no mayor que cero, cuando el valor absoluto del componente x del valor de diferencia de vector de movimiento es mayor que cero, codificar información representativa del valor absoluto del componente x del valor de diferencia de vector de movimiento, cuando el valor absoluto del componente y del valor de diferencia de vector de movimiento es mayor que cero, codificar información representativa del valor absoluto del componente y del valor de diferencia de vector de movimiento, cuando el valor absoluto del componente x del valor de diferencia de vector de movimiento es mayor que cero, codificar un signo del componente x del valor de diferencia de vector de movimiento y, cuando el valor absoluto del componente y del valor de diferencia de vector de movimiento es mayor que cero, codificar un signo del componente y del valor de diferencia de vector de movimiento. Por tanto, el procedimiento de la FIG. 9 representa otro ejemplo de un procedimiento para codificar conjuntamente datos de un valor de diferencia de vector de movimiento. Además, el descodificador de vídeo 30 también puede ser configurado para realizar un procedimiento de descodificación de vídeo, esencialmente similar al procedimiento de la FIG. 7. Ciertas técnicas de esta divulgación han sido evaluadas experimentalmente. Específicamente, las técnicas de esta divulgación referidas a la señalización adaptativa de precisión de sub-píxel fueron implementadas para HM3.0 y simuladas en condiciones comunes de prueba. Las prestaciones se comparan con un ancla de HM3.0 en términos de la Tasa de Bits del Delta de Bjøntegaard (BD). Los resultados de simulación mostraron un promedio del 0,4% de ahorro de la tasa BD sobre las configuraciones de alta eficacia, y un promedio del 0,6% de ahorro de la tasa BD sobre las configuraciones de baja complejidad. La Tabla 3 dada a continuación resume los resultados experimentales:

TABLA 3

	HE de acceso aleatorio			LC de acceso aleatorio		
	Y	U	V	Y	U	V
Clase A	0,0	-0,3	0,0	0,0	-0,4	-0,1
Clase B	-0,3	-0,3	-0,4	-0,3	-0,2	-0,2

ES 2 546 678 T3

Clase C	-0,5	-0,4	-0,3	-0,6	-0,7	-0,7
Clase D	-0,9	-0,7	-1,0	-1,3	-0,9	-1,0
Clase E						
Global	-0,4	-0,4	-0,4	-0,6	-0,6	-0,5
Tiempo de codificación [%]	110%			116%		
Tiempo de descodificación [%]	101%			102%		

	B HE de bajo retardo			B LC de bajo retardo		
	Y	U	V	Y	U	V
Clase A						
Clase B	-0,1	0,1	-0,1	-0,1	-0,4	-0,4
Clase C	-0,6	-0,6	-0,6	-0,8	-0,4	-0,5
Clase D	-1,2	-1,4	-1,7	-1,8	0,9	2,2
Clase E	0,4	-0,4	-0,8	0,7	1,1	0,5
Global	-0,4	-0,5	-0,8	-0,6	0,2	0,4
Tiempo de codificación [%]	110%			117%		
Tiempo de descodificación [%]	101%			103%		

5 Ha de reconocerse que, según el ejemplo, ciertos actos o sucesos de cualquiera de las técnicas descritas en la presente memoria pueden ser realizados en una secuencia distinta, pueden ser añadidos, combinados o descartados completamente (p. ej., no todos los actos o sucesos descritos son necesarios para la práctica de las técnicas). Además, en ciertos ejemplos, los actos o sucesos pueden ser realizados simultáneamente, p. ej., mediante procesamiento de múltiples hebras, procesamiento de interrupciones o múltiples procesadores, en lugar de secuencialmente.

10 En uno o más ejemplos, las funciones descritas pueden ser implementadas en hardware, software, firmware o cualquier combinación de los mismos. Si se implementan en software, las funciones pueden ser almacenadas en, o transmitidas por, una o más instrucciones o código en un medio legible por ordenador, y ejecutadas por una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a medios tangibles tales como medios de almacenamiento de datos, o medios de comunicación que incluyen a cualquier medio que facilite la transferencia de un programa de ordenador desde un lugar a otro, p. ej., de acuerdo a un protocolo de comunicación. De esta manera, los medios legibles por ordenador pueden
15 corresponder generalmente a (1) medios tangibles de almacenamiento legibles por ordenador, que no son transitorios o (2) un medio de comunicación tal como una señal u onda portadora. Los medios de almacenamiento de datos pueden ser medios disponibles cualesquiera que puedan ser objeto de acceso por parte de uno o más ordenadores, o uno o más procesadores, para extraer instrucciones, código y / o estructuras de datos para la implementación de las técnicas descritas en esta divulgación. Un producto de programa de ordenador puede incluir un medio legible por ordenador.

20 A modo de ejemplo, y no de limitación, tales medios de almacenamiento legibles por ordenador pueden comprender memorias 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 ser usado para almacenar el código de programa deseado en forma de instrucciones o estructuras de datos y que pueda ser objeto de acceso por parte de un ordenador. Además, cualquier conexión es debidamente denominada un medio legible por
25 ordenador. Por ejemplo, si las instrucciones son transmitidas desde una sede de la Red, un servidor u otro origen remoto usando un cable coaxial, cable de fibra óptica, un par cruzado, una línea de abonado digital (DSL) o tecnologías inalámbricas tales como los infrarrojos, la radio y las micro-ondas, entonces el cable coaxial, el cable de fibra óptica, el par cruzado, la DSL o las tecnologías inalámbricas tales como los infrarrojos, la radio y las micro-ondas están incluidos en la definición de medio. Debería entenderse, sin embargo, que los medios de almacenamiento legibles por ordenador y los
30 medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino

5 que se orientan, en cambio, a medios tangibles de almacenamiento no transitorio. Los discos, según se usan en la presente memoria, 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 usualmente los datos magnéticamente, mientras que otros discos reproducen los datos ópticamente con láseres. Las combinaciones de los anteriores también deberían ser incluidas dentro del ámbito de los medios legibles por ordenador.

10 Las instrucciones pueden ser ejecutadas por uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), formaciones de compuertas programables en el terreno (FPGA), u otros circuitos equivalentes, integrados o de lógica discreta. En consecuencia, el término "procesador", según se usa en la presente memoria, puede referirse a cualquiera de las estructuras precedentes, o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en la presente memoria. Además, en algunos aspectos, la funcionalidad descrita en la presente memoria puede ser proporcionada dentro de hardware dedicado y / o de módulos de software configurados para codificar y decodificar, o incorporados en un códec combinado. Además, las técnicas podrían ser totalmente implementadas en uno o más circuitos o elementos lógicos.

15 Las técnicas de esta divulgación pueden ser implementadas en una amplia variedad de dispositivos o aparatos, incluyendo un equipo de mano inalámbrico, un circuito integrado (IC) o un conjunto de IC (p. ej., un conjunto de chips). Diversos componentes, módulos o unidades se describen en esta divulgación para subrayar los aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no necesariamente requieren la realización por distintas unidades de hardware. En cambio, según lo descrito anteriormente, diversas unidades pueden ser combinadas
20 en una unidad de hardware de códec, o proporcionadas por una colección de unidades de hardware inter-operativas, incluyendo uno o más procesadores según lo descrito anteriormente, conjuntamente con el software y / o firmware adecuado.

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

REIVINDICACIONES

1. Un procedimiento (220; 320) de codificación por entropía de datos de vídeo, comprendiendo el procedimiento:

5 cuando un valor absoluto de un componente x de un valor de diferencia de vector de movimiento para un bloque actual es mayor que cero (226; 326), y cuando un valor absoluto de un componente y del valor de diferencia de vector de movimiento para el bloque actual es mayor que cero (228; 328), codificar por entropía información representativa de un valor de diferencia de vector de movimiento, en el que la codificación por entropía de la información representativa del valor de diferencia de vector de movimiento comprende:

10 intercalar información representativa de si el valor absoluto del componente x de un valor de diferencia de vector de movimiento para el bloque actual de datos de vídeo es o no mayor que cero, e información representativa de si el valor absoluto del componente y del valor de diferencia de vector de movimiento es o no mayor que cero; e
intercalar información representativa del valor absoluto del componente x del valor de diferencia de vector de movimiento, información representativa del valor absoluto del componente y del valor de diferencia de vector de movimiento,
15 un signo del componente x del valor de diferencia de vector de movimiento (230; 330) y un signo del componente y del valor de diferencia de vector de movimiento (232; 332).

2. El procedimiento de la reivindicación 1, en el que la codificación por entropía comprende la codificación aritmética binaria adaptativa según el contexto, CABAC, comprendiendo además el procedimiento:

20 calcular un vector de movimiento para el bloque actual de datos de vídeo; y
calcular (156) el valor de diferencia de vector de movimiento como una diferencia entre el vector de movimiento para el bloque actual de datos de vídeo y un predictor de vector de movimiento, determinado para el bloque actual de datos de vídeo.

3. El procedimiento de la reivindicación 1, en el que la codificación por entropía comprende la codificación de la codificación aritmética binaria adaptativa según el contexto, CABAC, comprendiendo además el procedimiento:

25 determinar (286) un predictor de vector de movimiento para el bloque actual de datos de vídeo; y
calcular (288) un vector de movimiento para el bloque actual de datos de vídeo como una suma del valor de diferencia de vector de movimiento y el predictor de vector de movimiento.

4. El procedimiento de la reivindicación 1, que comprende además:

30 codificar un valor representativo de si un vector de movimiento del bloque actual de datos de vídeo tiene una primera precisión de sub-píxel o una segunda precisión de sub-píxel, en el que la segunda precisión de sub-píxel es mayor que la primera precisión de sub-píxel; y
cuando el vector de movimiento tiene la segunda precisión de sub-píxel, cuando el componente x del valor de diferencia de vector de movimiento no es igual a cero, y cuando el componente y del valor de diferencia de vector de movimiento no es igual a cero,
35 en el que, la codificación de información representativa del valor absoluto del componente x del valor de diferencia de vector de movimiento y la codificación de información representativa del valor absoluto del componente y del valor de diferencia de vector de movimiento comprenden codificar un valor conjuntamente codificado, representativo tanto del componente x del valor de diferencia de vector de movimiento como del componente y del valor de diferencia de vector de movimiento.

40 5. El procedimiento de la reivindicación 4, que comprende además determinar si el vector de movimiento tiene la primera precisión de sub-píxel o la segunda precisión de sub-píxel, y que comprende:

determinar un valor de umbral asociado al vector de movimiento; y
determinar que el vector de movimiento tiene la primera precisión de sub-píxel cuando tanto el componente x del valor de diferencia de vector de movimiento como el componente y del valor de diferencia de vector de movimiento
45 son mayores que el umbral.

6. El procedimiento de la reivindicación 5, en el que la determinación de si el vector de movimiento tiene la primera precisión de sub-píxel o la segunda precisión de sub-píxel comprende, cuando al menos uno entre el componente x del valor de diferencia de vector de movimiento y el componente y del valor de diferencia de vector de movimiento es menor o igual que el umbral, determinar si el vector de movimiento tiene la primera precisión de sub-píxel o la segunda precisión de sub-píxel, en base a un valor de un indicador de resolución de vector de movimiento.
50

7. El procedimiento de la reivindicación 1,

en el que la codificación de información representativa del valor absoluto del componente x de los valores de diferencia de vectores de movimiento comprende codificar el valor absoluto del componente x de los valores de diferencia de vectores de movimiento como $\text{mvd}_x/2 - 1$, en el que mvd_x comprende el componente x del valor de diferencia de vector de movimiento; y

5 en el que la codificación de información representativa del valor absoluto del componente y de los valores de diferencia de vectores de movimiento comprende codificar el valor absoluto del componente y de los valores de diferencia de vectores de movimiento como $\text{mvd}_y/2 - 1$, en el que mvd_y comprende el componente y del valor de diferencia de vector de movimiento.

8. Un aparato (56; 70) para codificar por entropía datos de vídeo, comprendiendo el aparato:

10 medios para codificar por entropía información representativa de un valor de diferencia de vector de movimiento cuando el valor absoluto del componente x del valor de diferencia de vector de movimiento es mayor que cero y cuando el valor absoluto del componente y del valor de diferencia de vector de movimiento es mayor que cero, en el que el medio para codificar por entropía la información representativa del valor de diferencia de vector de movimiento comprende:

15 medios para intercalar información representativa de si un valor absoluto de un componente x de un valor de diferencia de vector de movimiento para un bloque actual de datos de vídeo es o no mayor que cero, e información representativa de si un valor absoluto de un componente y del valor de diferencia de vector de movimiento es o no mayor que cero;

20 medios para intercalar información representativa del valor absoluto del componente x del valor de diferencia de vector de movimiento, información representativa del valor absoluto del componente y del valor de diferencia de vector de movimiento, un signo del componente x del valor de diferencia de vector de movimiento, y un signo del componente y del valor de diferencia de vector de movimiento.

25 9. El aparato de la reivindicación 8, en el que el medio para la codificación por entropía comprende medios para la codificación de la codificación aritmética binaria adaptativa según el contexto, CABAC, comprendiendo además el aparato:

30 medios para calcular un vector de movimiento para el bloque actual de datos de vídeo; y
medios para calcular el valor de diferencia de vector de movimiento como una diferencia entre el vector de movimiento para el bloque actual de datos de vídeo y un predictor de vector de movimiento determinado para el bloque actual de datos de vídeo.

10. El aparato de la reivindicación 8, en el que el medio para la codificación por entropía comprende medios para la decodificación de la codificación aritmética binaria adaptativa según el contexto, CABAC, comprendiendo además el aparato:

35 medios para determinar un predictor de vector de movimiento para el bloque actual de datos de vídeo; y
medios para calcular un vector de movimiento para el bloque actual de datos de vídeo como una suma del valor de diferencia de vector de movimiento y el predictor de vector de movimiento.

11. El aparato de la reivindicación 8, que comprende además:

40 medios para codificar un valor representativo de si un vector de movimiento del bloque actual de datos de vídeo tiene una primera precisión de sub-píxel o una segunda precisión de sub-píxel, en el que la segunda precisión de sub-píxel es mayor que la primera precisión de sub-píxel; y

45 medios para codificar un valor conjuntamente codificado, representativo tanto del componente x del valor de diferencia de vector de movimiento como del componente y del valor de diferencia de vector de movimiento, cuando el vector de movimiento tiene la segunda precisión de sub-píxel, cuando el componente x del valor de diferencia de vector de movimiento no es igual a cero, y cuando el componente y del valor de diferencia de vector de movimiento no es igual a cero.

12. El aparato de la reivindicación 11, que comprende además medios para determinar si el vector de movimiento tiene la primera precisión de sub-píxel o la segunda precisión de sub-píxel, y que comprende:

50 medios para determinar un valor de umbral asociado al vector de movimiento; y
medios para determinar que el vector de movimiento tiene la primera precisión de sub-píxel cuando tanto el componente x del valor de diferencia de vector de movimiento como el componente y del valor de diferencia del vector de movimiento son mayores que el umbral.

13. El aparato de la reivindicación 12, en el que el medio para determinar si el vector de movimiento tiene la primera

precisión de sub-píxel o la segunda precisión de sub-píxel comprende medios para determinar si el vector de movimiento tiene la primera precisión de sub-píxel o la segunda precisión de sub-píxel, en base a un valor de un indicador de resolución de vector de movimiento, cuando al menos uno entre el componente x del valor de diferencia de vector de movimiento y el componente y del valor de diferencia del vector de movimiento es menor o igual que el umbral.

5 14. El aparato de la reivindicación 8,

en el que el medio para codificar información representativa del valor absoluto del componente x de los valores de diferencia de vectores de movimiento comprende medios para codificar el valor absoluto del componente x de los valores de diferencia de vectores de movimiento como $mvd_x/2 - 1$, en el que mvd_x comprende el componente x del valor de diferencia de vector de movimiento; y

10 en el que el medio para codificar información representativa del valor absoluto del componente y de los valores de diferencia de vectores de movimiento comprende medios para codificar el valor absoluto del componente y de los valores de diferencia de vectores de movimiento como $mvd_y/2 - 1$, en el que mvd_y comprende el componente y del valor de diferencia de vector de movimiento.

15 15. Un producto de programa de ordenador que comprende un medio de almacenamiento legible por ordenador, que tiene almacenadas en el mismo instrucciones que, cuando son ejecutadas, provocan que uno o más procesadores de un dispositivo para codificar datos de vídeo lleven a cabo el procedimiento de cualquiera de las reivindicaciones 1 a 7.

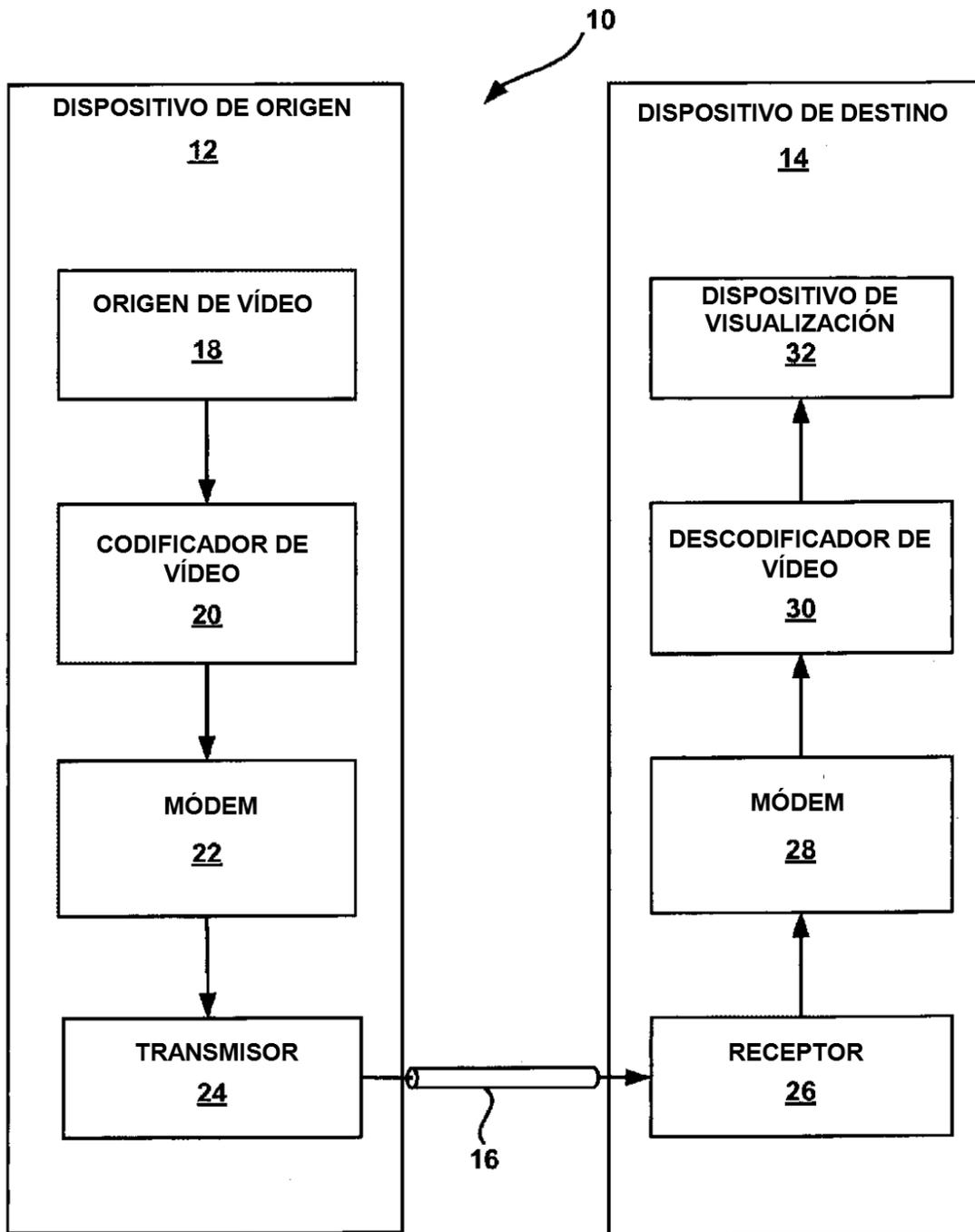


FIG. 1

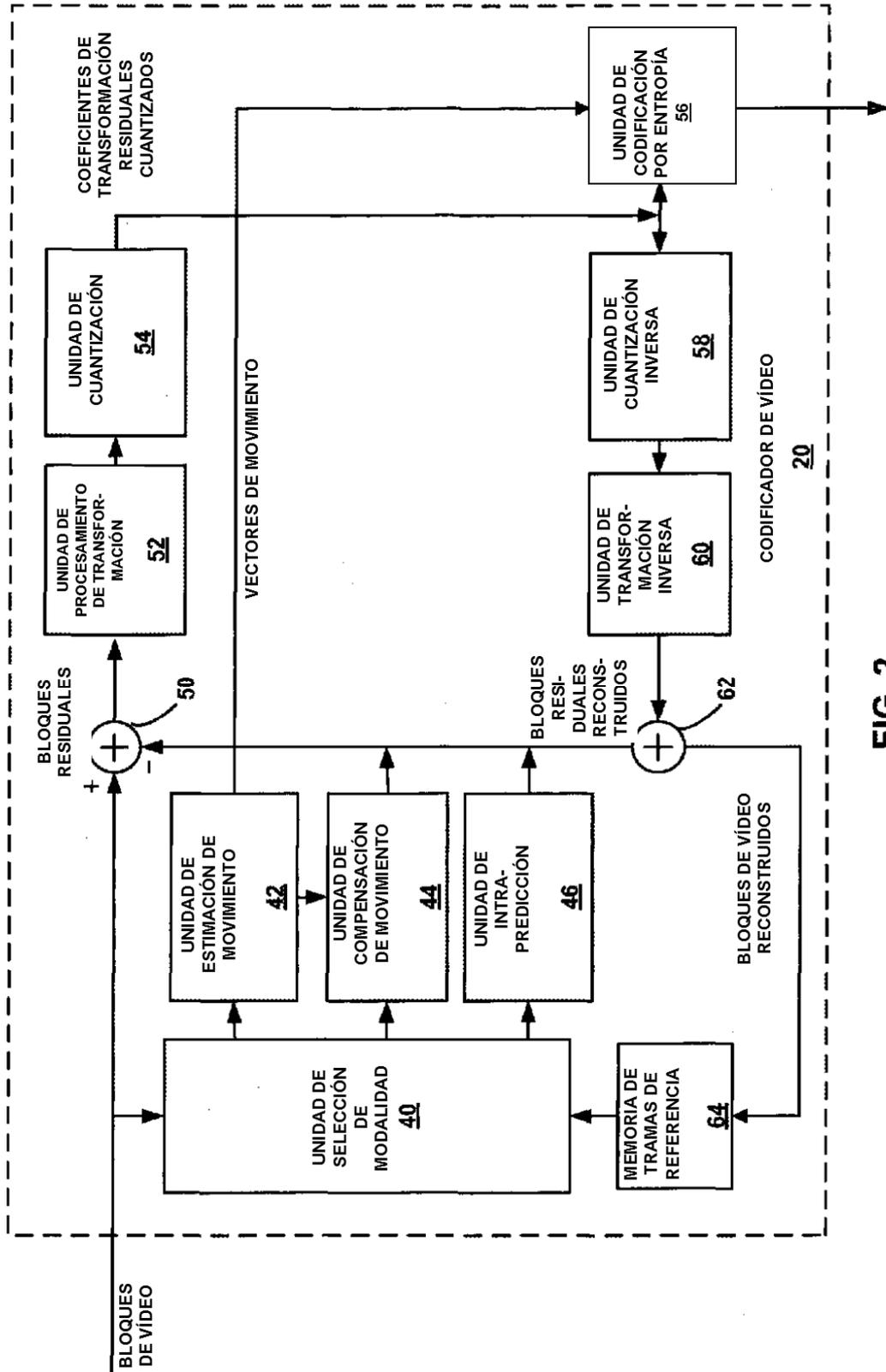


FIG. 2

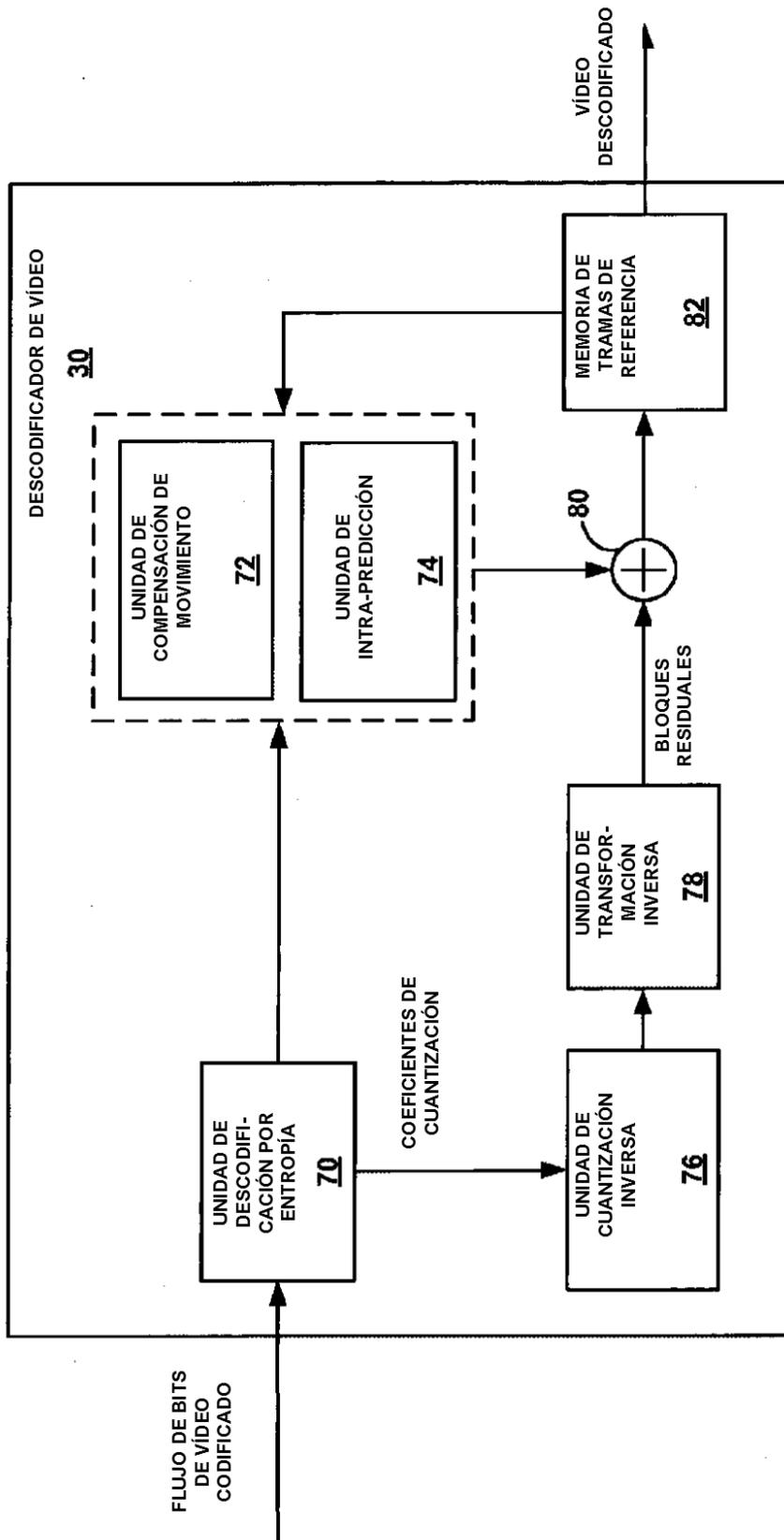


FIG. 3

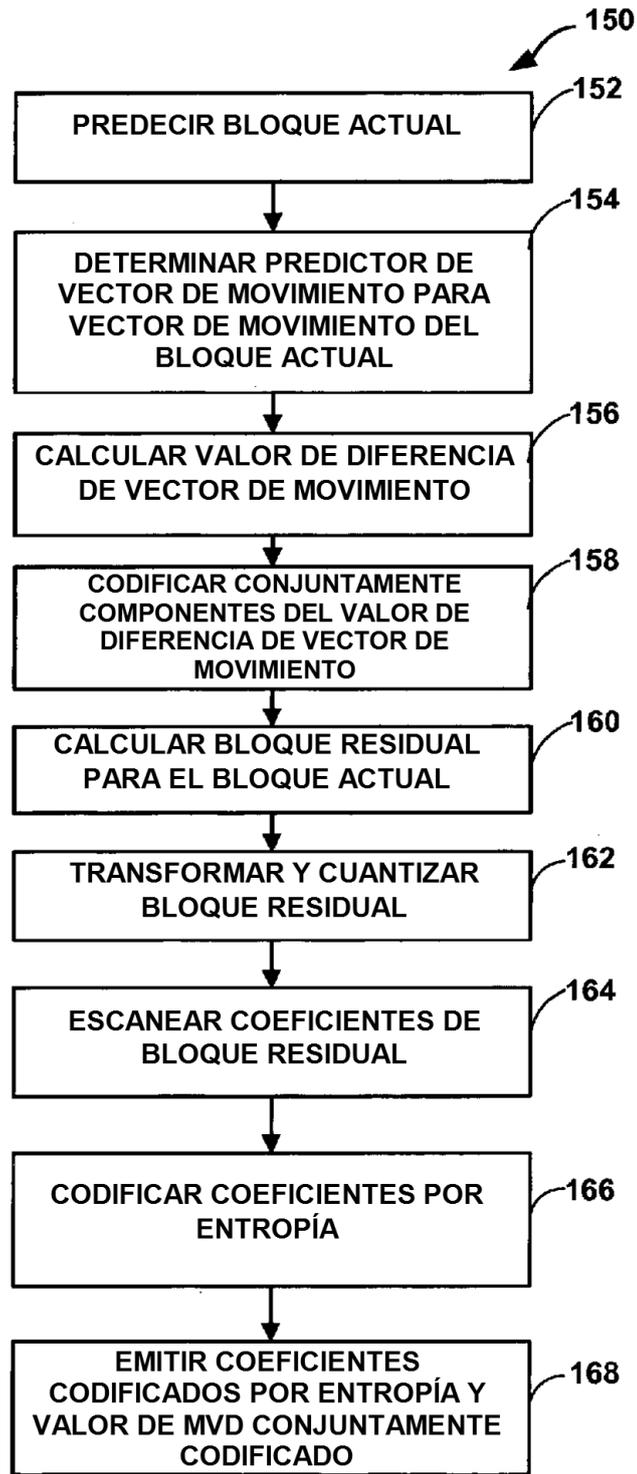


FIG. 4

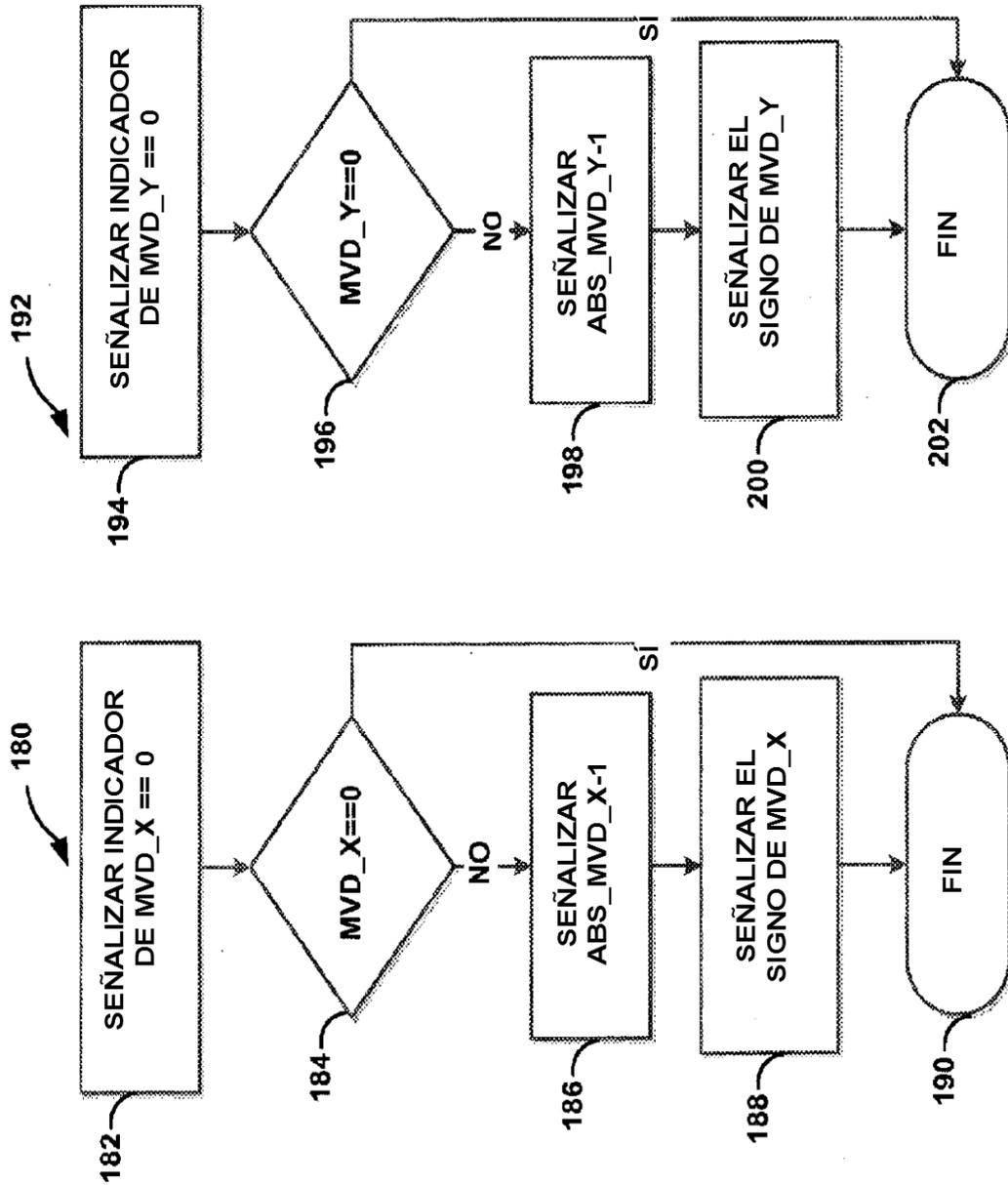


FIG. 5

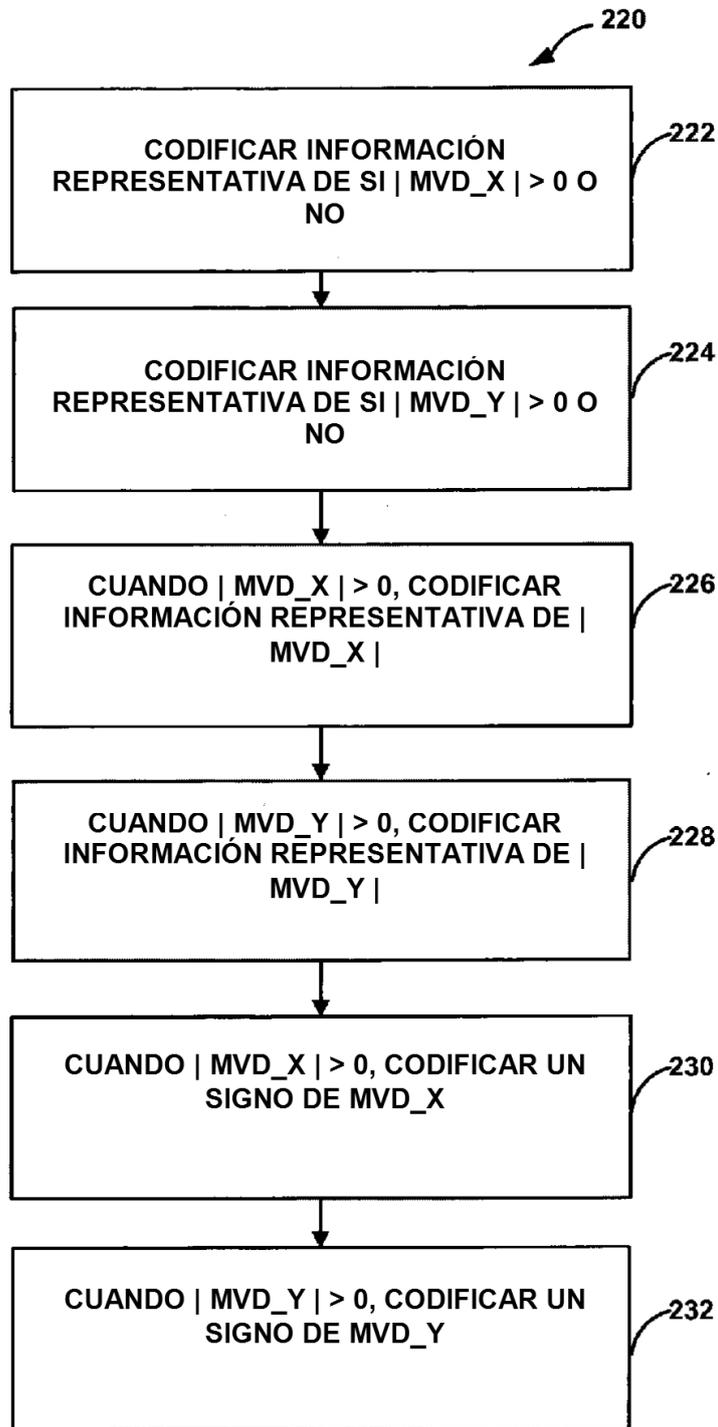


FIG. 6

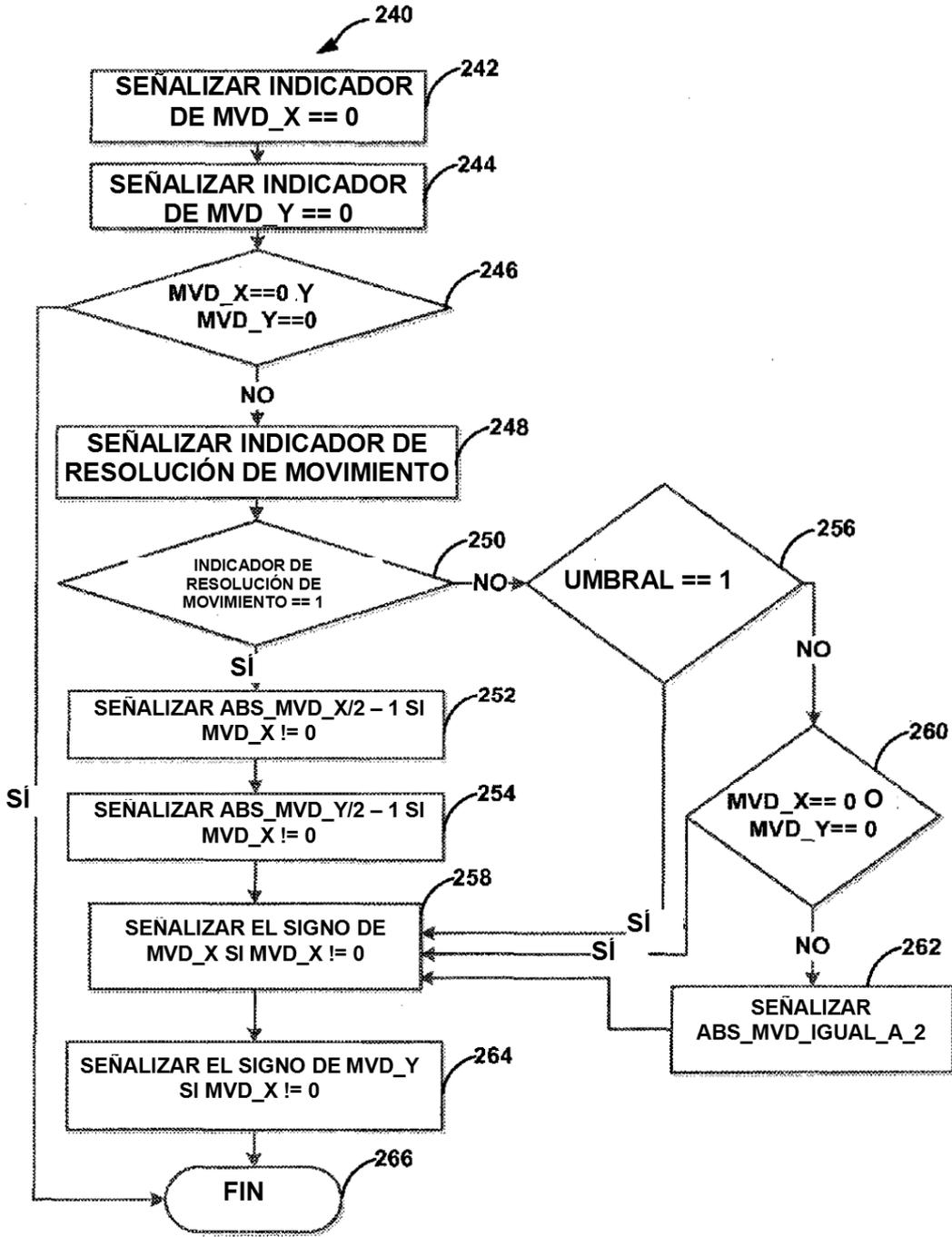


FIG. 7

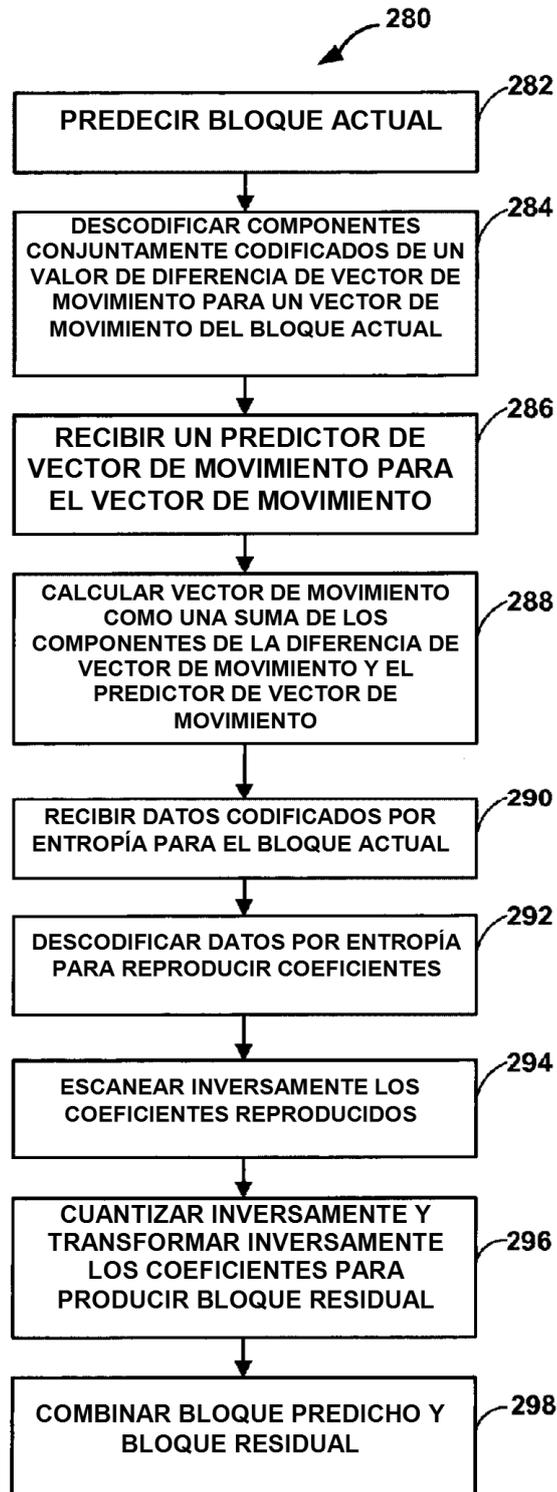


FIG. 8

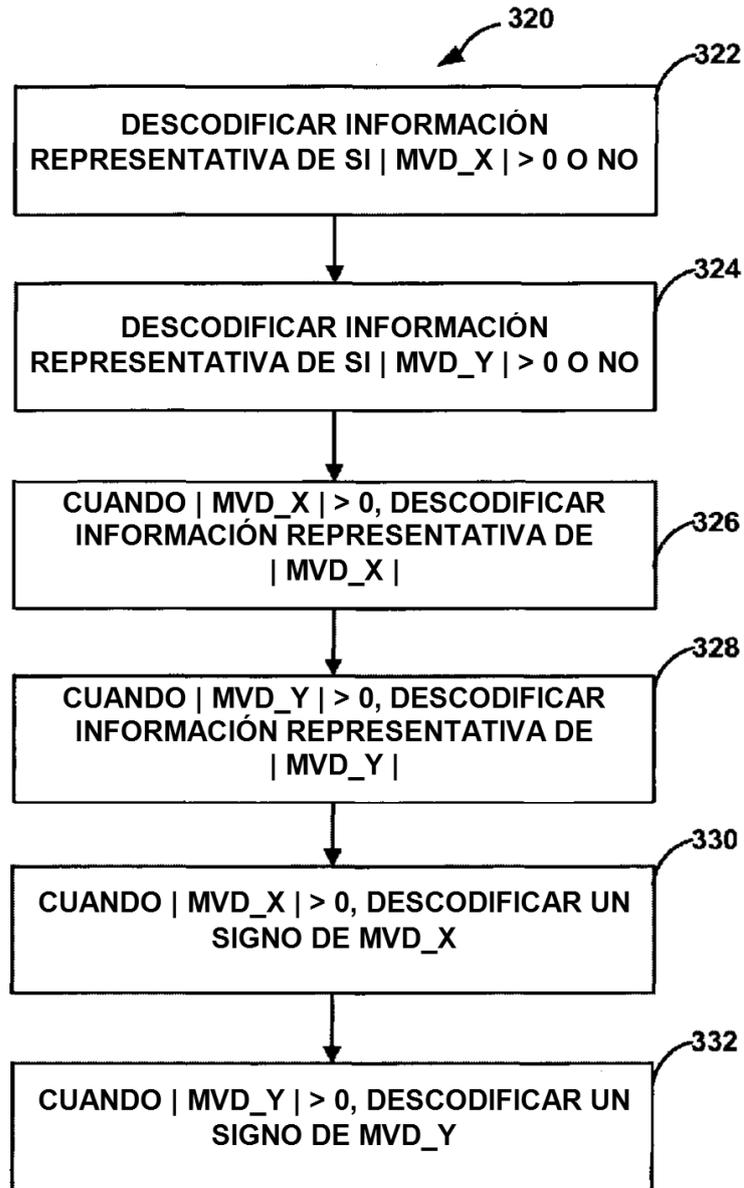


FIG. 9