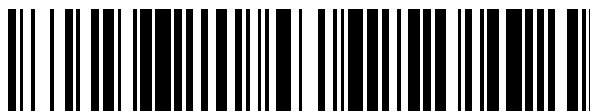


19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 880 744**

51 Int. Cl.:

**H04N 19/44** (2014.01)  
**H04N 19/51** (2014.01)  
**H04N 19/56** (2014.01)  
**H04N 19/53** (2014.01)  
**H04N 19/523** (2014.01)  
**H04N 19/176** (2014.01)

12

## TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **09.01.2015 PCT/US2015/010852**
- 87 Fecha y número de publicación internacional: **16.07.2015 WO15106126**
- 96 Fecha de presentación y número de la solicitud europea: **09.01.2015 E 15702038 (9)**
- 97 Fecha y número de publicación de la concesión europea: **30.06.2021 EP 3092801**

54 Título: **Señalización de resolución adaptable del vector de movimiento para la codificación de vídeo**

30 Prioridad:

**09.01.2014 US 201461925633 P**  
**17.03.2014 US 201461954457 P**  
**16.10.2014 US 201462064761 P**  
**08.01.2015 US 201514592767**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:  
**25.11.2021**

73 Titular/es:

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

72 Inventor/es:

**LI, XIANG;**  
**SOLE ROJALS, JOEL;**  
**KARCZEWICZ, MARTA;**  
**LIU, HONGBIN;**  
**ZHANG, LI y**  
**CHEN, JIANLE**

74 Agente/Representante:

**ISERN JARA, Jorge**

ES 2 880 744 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

Señalización de resolución adaptable del vector de movimiento para la codificación de vídeo

## 5 CAMPO TÉCNICO

Esta divulgación se refiere a la codificación de vídeo y, más particularmente, a la codificación de vídeo por intrapredicción.

## 10 ANTECEDENTES

Las capacidades de vídeo digital se pueden incorporar a una amplia gama de dispositivos, que incluye televisores digitales, sistemas de radiodifusión directa digital, sistemas de radiodifusión inalámbrica, asistentes personales digitales (PDA), ordenadores portátiles o de escritorio, cámaras digitales, dispositivos de grabación digitales, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos de radio celulares o por satélite, dispositivos de videoconferencia y similares. Los dispositivos de vídeo digitales implementan técnicas de compresión de vídeo, tales como las descritas en las normas definidas por MPEG-2, MPEG-4, ITU-T H.263 o ITU-T H.264/MPEG-4, parte 10, Codificación de Vídeo Avanzada (AVC) y ampliaciones de dichas normas, para transmitir y recibir información de vídeo digital más eficazmente.

Las técnicas de compresión de vídeo realizan predicción espacial y/o predicción temporal para reducir o eliminar la redundancia inherente a las secuencias de vídeo. Para la codificación de vídeo basada en bloques, una trama o un fragmento de vídeo se pueden dividir en macrobloques. Cada macrobloque se puede dividir aún más. Los macrobloques de una trama o un fragmento intracodificados (I) se codifican usando predicción espacial con respecto a macrobloques contiguos. Los macrobloques de una trama o un fragmento intercodificados (P o B) pueden usar predicción espacial con respecto a macrobloques contiguos de la misma trama o fragmento, o predicción temporal con respecto a otras tramas de referencia.

Y-J CHIU *ET AL.*, "CE1: Report of self-derivation of motion estimation in TMuC 0.9", 95. REUNIÓN MPEG; 24-1-2011 - 28-1-2011; DAEGU; (GRUPO DE EXPERTOS DE IMÁGENES EN MOVIMIENTO O ISO/IEC JTC1/SC29/WG11), (20110121), n.º m18919 describe SDME basado en candidatos redondeados con y sin refinamiento de MV.

## SUMARIO

En general, esta divulgación describe técnicas para seleccionar de forma adaptable la precisión del vector de movimiento para los vectores de movimiento que se utilizan para codificar bloques de datos de vídeo y para determinar mediante un decodificador de vídeo la misma precisión del vector de movimiento que el codificador de vídeo ha seleccionado para cada uno de los bloques.

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

## BREVE DESCRIPCIÓN DE LOS DIBUJOS

La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y decodificación de vídeo de ejemplo que puede utilizar las técnicas de esta divulgación para dar soporte a la resolución adaptable del vector de movimiento.

La FIG. 2 es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo que puede implementar técnicas para dar soporte a la resolución adaptable del vector de movimiento.

La FIG. 3 es un diagrama de bloques que ilustra un ejemplo de un decodificador de vídeo, que decodifica una secuencia de vídeo codificada.

La FIG. 4 es un diagrama conceptual que ilustra posiciones de píxeles fraccionarios para una posición de píxel completo.

Las FIG. 5A-5C son diagramas conceptuales que ilustran las correspondientes posiciones de píxeles de crominancia y luminancia.

La FIG. 6 es una ilustración de un ejemplo de coincidencia de la plantilla en forma de L para la derivación del vector de movimiento en el lado del decodificador (DMVD).

La FIG. 7 es un diagrama conceptual que ilustra un ejemplo de derivación del MV bidireccional basada en espejos.

La FIG. 8 es un diagrama de flujo que ilustra un procedimiento de ejemplo para codificar datos de vídeo de acuerdo con las técnicas descritas en esta divulgación.

La FIG. 9 es un diagrama de flujo que ilustra un procedimiento de ejemplo para decodificar datos de vídeo de acuerdo con las técnicas descritas en esta divulgación.

La FIG. 10 es un diagrama de flujo que ilustra un procedimiento de ejemplo para codificar datos de vídeo.

La FIG. 11 es un diagrama de flujo que ilustra un procedimiento de ejemplo para codificar datos de vídeo.

La FIG. 12 es un diagrama de flujo que ilustra un procedimiento de ejemplo para codificar datos de vídeo.

La FIG. 13 es un diagrama de flujo que ilustra un procedimiento de ejemplo para codificar datos de vídeo.  
 La FIG. 14 es un diagrama de flujo que ilustra un procedimiento de ejemplo para codificar datos de vídeo.  
 La FIG. 15 es un diagrama de flujo que ilustra un procedimiento de ejemplo para codificar datos de vídeo.  
 La FIG. 16 es un diagrama de flujo que ilustra un procedimiento de ejemplo para codificar datos de vídeo.  
 La FIG. 17 es un diagrama de flujo que ilustra un procedimiento de ejemplo para codificar datos de vídeo.

#### DESCRIPCIÓN DETALLADA

En general, esta divulgación describe técnicas para seleccionar de forma adaptable la precisión del vector de movimiento para los vectores de movimiento que se utilizan para codificar bloques de datos de vídeo y para determinar mediante un descodificador de vídeo la misma precisión del vector de movimiento que la que ha sido seleccionada por el codificador de vídeo. De acuerdo con algunas técnicas, el descodificador de vídeo puede derivar, sin señalización explícita en el flujo de bits de vídeo codificado, la precisión del vector de movimiento seleccionada por el codificador de vídeo. De acuerdo con otras técnicas, el codificador de vídeo puede señalar, en el flujo de bits de vídeo codificado, la precisión del vector de movimiento seleccionada. Las técnicas de esta divulgación pueden incluir, por ejemplo, seleccionar de forma adaptable entre la precisión de un píxel entero y diferentes niveles de precisión de un píxel subentero, a veces denominada precisión de píxel fraccionario. Por ejemplo, las técnicas pueden incluir la selección adaptable entre la precisión de un píxel entero y precisión de un cuarto de píxel o precisión de un octavo de píxel para los vectores de movimiento utilizados para codificar bloques de datos de vídeo. El término precisión de "octavo de píxel" en esta divulgación está destinado a referirse a la precisión de un octavo ( $1/8$ ) de un píxel, por ejemplo, uno de: la posición de un píxel completo (0/8), un octavo de un píxel (1/8), dos octavos de un píxel (2/8, también un cuarto de un píxel), tres octavos de un píxel (3/8), cuatro octavos de un píxel (4/8, también la mitad de un píxel y dos cuartos de un píxel), cinco octavos de un píxel (5/8), seis octavos de un píxel (6/8, también tres cuartos de un píxel), o siete octavos de un píxel (7/8).

Los codificadores y descodificadores H.264 y H.265 convencionales admiten vectores de movimiento con una precisión de un cuarto de píxel. Sin embargo, dicha precisión de píxeles no se señala ni se deriva, sino que es fija. En algunos casos, la precisión de un octavo de píxel puede proporcionar determinadas ventajas sobre la precisión de un cuarto de píxel o la precisión de un píxel entero. Sin embargo, la codificación de cada vector de movimiento con una precisión de un octavo de píxel puede requerir demasiados bits de codificación que pueden superar los beneficios que proporcionan los vectores de movimiento con precisión de un octavo de píxel. Para algunos tipos de contenido de vídeo, puede ser preferente codificar vectores de movimiento sin interpolación en absoluto, en otras palabras, usando solo precisión de un píxel entero.

El contenido de la pantalla como, por ejemplo, el contenido generado por un ordenador típicamente involucra a una serie de píxeles que tienen exactamente los mismos valores de píxeles, seguidos de un cambio brusco en los valores de los píxeles. Por ejemplo, en el contenido de la pantalla que incluye texto azul sobre un fondo blanco, los píxeles que forman una letra azul pueden tener todos los mismos valores de píxeles, mientras que el fondo blanco también tiene todos los mismos valores de píxeles, pero los valores de los píxeles blancos pueden ser significativamente diferentes que los valores de los píxeles azules. El contenido adquirido por la cámara, por el contrario, típicamente incluye cambios lentos en los valores de los píxeles como consecuencia del movimiento, las sombras, los cambios de iluminación y otros fenómenos naturales. Dado que el contenido de la pantalla y el contenido adquirido por la cámara típicamente tienen características diferentes, las herramientas de codificación que son eficaces para uno pueden no serlo necesariamente para el otro. Para poner un ejemplo, la interpolación de subpíxeles para la codificación por interpredicción puede mejorar la codificación del contenido de la cámara, pero la complejidad asociada y la sobrecarga de señalización pueden reducir en realidad la calidad de la codificación y/o la eficacia del ancho de banda para el contenido de la pantalla.

Las técnicas de esta divulgación incluyen determinar de forma adaptable la precisión del vector de movimiento en base a, por ejemplo, el contenido del vídeo que se codifica. En algunos ejemplos, las técnicas de esta divulgación incluyen derivar, mediante un codificador, una precisión del vector de movimiento adecuada para el contenido de vídeo que se codifica. Usando las mismas técnicas de derivación, un descodificador de vídeo también puede determinar, sin recibir un elemento de sintaxis que indique la precisión del vector de movimiento, qué precisión del vector de movimiento se ha utilizado para codificar los datos de vídeo. En otros ejemplos, un codificador de vídeo puede señalar, en el flujo de bits de vídeo codificado, la precisión del vector de movimiento seleccionada por el codificador de vídeo.

Seleccionar de forma adaptable la precisión del vector de movimiento puede mejorar la calidad global de la codificación de vídeo al permitir que los vectores de movimiento con precisión más alta (por ejemplo, vectores de movimiento con precisión de  $1/4$  o  $1/8$ ) se utilicen para el contenido de vídeo en los que el uso de dichos vectores de movimiento de alta precisión mejora la calidad de la codificación de vídeo, por ejemplo, produciendo una mejor compensación entre velocidad y distorsión. La selección adaptable de la precisión del vector de movimiento también puede mejorar la calidad global de la codificación de vídeo al permitir el uso de vectores de movimiento de menor precisión (por ejemplo, precisión de número entero) para el contenido de vídeo en el que el uso de vectores de movimiento con mayor precisión no mejora, o incluso empeora, la calidad de la codificación de vídeo.

Diversas técnicas en esta divulgación pueden describirse en referencia a un codificador de vídeo, que está destinado a ser un término genérico que puede referirse a un codificador de vídeo o un descodificador de vídeo. A menos que se indique explícitamente lo contrario, no se debe suponer que las técnicas descritas con respecto a un codificador de vídeo o un descodificador de vídeo no pueden ser realizadas por el otro de un codificador de vídeo o un descodificador de vídeo. Por ejemplo, en muchos casos, un descodificador de vídeo realiza la misma técnica de codificación, o en ocasiones recíproca, que un codificador de vídeo para descodificar datos de vídeo codificados. En muchos casos, un codificador de vídeo también incluye un bucle de descodificación de vídeo y, por tanto, el codificador de vídeo realiza la descodificación de vídeo como parte de la codificación de datos de vídeo. Por tanto, a menos que se establezca de otro modo, las técnicas descritas en esta divulgación con respecto a un descodificador de vídeo también pueden ser realizadas por un codificador de vídeo, y viceversa.

Esta divulgación también puede utilizar términos como capa actual, bloque actual, imagen actual, fragmento actual, etc. En el contexto de esta divulgación, el término actual está destinado a identificar una capa, bloque, imagen, fragmento, etc. que se codifica actualmente, a diferencia de, por ejemplo, capas, bloques, imágenes y fragmentos previamente codificados o bloques, imágenes y fragmentos aún por codificar.

La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y descodificación de vídeo de ejemplo 10 que puede utilizar las técnicas de esta divulgación para dar soporte a la resolución adaptable del vector de movimiento. Como se muestra en la FIG. 1, el sistema 10 incluye un dispositivo de origen 12 que transmite el vídeo codificado a un dispositivo de destino 14 por medio de un canal de comunicación 16. El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender cualquiera de 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 teléfonos inalámbricos, los denominados radioteléfonos celulares o vía satélite, o cualquier dispositivo inalámbrico que pueda comunicar información de vídeo a través de un canal de comunicación 16, en cuyo caso el canal de comunicación 16 es inalámbrico. Sin embargo, las técnicas de esta divulgación, que en general se refieren a técnicas para dar soporte a la precisión de subpíxel adaptable, no están necesariamente limitadas a aplicaciones o configuraciones inalámbricas. Por ejemplo, estas técnicas se pueden aplicar a radiodifusiones de televisión por aire, a transmisiones de televisión por cable, a transmisiones de televisión vía satélite, a transmisiones de vídeo por Internet, a vídeo digital codificado que se codifica en un medio de almacenamiento, o a otros escenarios. En consecuencia, el canal de comunicación 16 puede comprender cualquier combinación de medios inalámbricos o alámbricos adecuados para la transmisión de datos de vídeo codificados.

En el ejemplo de la FIG. 1, el dispositivo de origen 12 incluye una fuente de vídeo 18, un codificador de vídeo 20, un modulador/desmodulador (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 con esta divulgación, el codificador de vídeo 20 del dispositivo de origen 12 puede estar configurado para aplicar las técnicas para dar soporte a la precisión de subpíxel adaptable en los vectores de movimiento. En otros ejemplos, un dispositivo de origen y un dispositivo de destino pueden incluir otros componentes o disposiciones. Por ejemplo, el dispositivo de origen 12 puede recibir datos de vídeo desde una fuente de vídeo externa 18, tal como una cámara externa. Del mismo modo, el dispositivo de destino 14 puede interactuar con un dispositivo de visualización externo, en lugar de incluir un dispositivo de visualización integrado.

El sistema 10 ilustrado de la FIG. 1 es meramente un ejemplo. Las técnicas para dar soporte a la precisión de subpíxel adaptable en los vectores de movimiento pueden realizarse mediante cualquier dispositivo de codificación y/o descodificación de vídeo digital. Aunque, en general, las técnicas de esta divulgación se realizan mediante un dispositivo de codificación de vídeo, las técnicas también se pueden realizar mediante un codificador/descodificador de vídeo, denominado típicamente "CÓDEC". Además, las técnicas de esta divulgación también se pueden realizar mediante un preprocesador de vídeo. El dispositivo de origen 12 y el dispositivo de destino 14 son meramente ejemplos de dichos dispositivos de codificación, donde el dispositivo de origen 12 genera datos de vídeo codificados para su transmisión al dispositivo de destino 14. En algunos ejemplos, los dispositivos 12, 14 pueden funcionar de manera sustancialmente simétrica, de modo que cada uno de los dispositivos 12, 14 incluye componentes de codificación y descodificación de vídeo. Por consiguiente, el sistema 10 puede admitir una transmisión de vídeo unidireccional o bidireccional entre los dispositivos de vídeo 12, 14, por ejemplo, para la transmisión continua de vídeo, reproducción de vídeo, radiodifusión de vídeo o videoteléfono.

La fuente de vídeo 18 del dispositivo de origen 12 puede incluir un dispositivo de captura de vídeo, tal como una cámara de vídeo, un archivo de vídeo que contiene vídeo capturado previamente y/o una transmisión de vídeo de un proveedor de contenido de vídeo. Como otra alternativa, la fuente de vídeo 18 puede generar datos basados en gráficos de ordenador como fuente de vídeo, o una combinación de vídeo en directo, vídeo archivado y vídeo generado por ordenador. En algunos casos, si la fuente de vídeo 18 es una cámara de vídeo, el dispositivo de origen 12 y el dispositivo de destino 14 pueden formar los denominados teléfonos con cámara o videoteléfonos. Sin embargo, como se menciona anteriormente, las técnicas descritas en esta divulgación pueden ser aplicables a la codificación de vídeo en general, y se pueden aplicar a aplicaciones inalámbricas y/o alámbricas. En cada caso, el codificador de vídeo 20 puede codificar el vídeo capturado, precapturado o generado por ordenador. La información de vídeo codificada se puede modular a continuación por el módem 22, de acuerdo con una norma de comunicación, y transmitirse al dispositivo de destino 14 por medio del 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.

El receptor 26 del dispositivo de destino 14 recibe información a través del canal 16, y el módem 28 desmodula la información. De nuevo, el proceso de codificación de vídeo puede implementar una o más de las técnicas descritas en el presente documento para dar soporte a la precisión de subpíxel adaptable en los vectores de movimiento. La información comunicada a través del canal 16 puede incluir información de sintaxis definida por el codificador de vídeo 20, que también es usada por el descodificador de vídeo 30, que incluye elementos de sintaxis que describen características y/o el procesamiento de macrobloques u otras unidades codificadas, por ejemplo, grupos de imágenes (GOP). El dispositivo de visualización 32 muestra los datos de vídeo descodificados a un usuario, y puede comprender cualquiera de una variedad de dispositivos de visualización, tales como un tubo de rayos catódicos (CRT), una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de visualización.

En el ejemplo de la FIG. 1, el canal de comunicación 16 puede comprender cualquier medio de comunicación inalámbrico o alámbrico, tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión físicas, o cualquier combinación de medios inalámbricos y alámbricos. 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 en general cualquier medio de comunicación adecuado, o un conjunto de medios de comunicación diferentes, para transmitir datos de vídeo desde el dispositivo de origen 12 hasta el dispositivo de destino 14, que incluye cualquier combinación adecuada de medios alámbricos 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 hasta el dispositivo de destino 14.

El codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con una norma de compresión de vídeo, tal como la norma ITU-T H.264, denominada de forma alternativa MPEG-4, parte 10, codificación avanzada de vídeo (AVC). Sin embargo, las técnicas de esta divulgación no se limitan a ninguna norma de codificación particular. Otros ejemplos incluyen MPEG-2 e ITU-T H.263. Aunque no se muestra en la FIG. 1, en algunos aspectos, el codificador de vídeo 20 y el descodificador de vídeo 30 se pueden integrar cada uno con un codificador y un descodificador de audio, y pueden incluir unidades MUX-DEMUX apropiadas, u otro hardware y software, para gestionar la codificación tanto de audio como de vídeo en un flujo de datos común o en flujos de datos separados. Cuando proceda, las unidades MUX-DEMUX pueden ajustarse al protocolo de multiplexor ITU H.223 u otros protocolos tales como el protocolo de datagramas de usuario (UDP).

La norma ITU-T H.264/MPEG-4 (AVC) se formuló por el Grupo de Expertos en Codificación de Vídeo (VCEG) de ITU-T conjuntamente con el Grupo de Expertos en Imágenes en Movimiento (MPEG) de ISO/IEC, como el producto de una asociación colectiva conocida como el Equipo Mixto de Vídeo (JVT). En algunos aspectos, las técnicas descritas en esta divulgación se pueden aplicar a dispositivos que se ajustan, en general, a la norma H.264. La norma H.264 está descrita en la recomendación H.264 de la ITU-T, codificación avanzada de vídeo para servicios audiovisuales genéricos, por el grupo de estudio de la ITU-T, y con fecha de marzo de 2005, que se puede denominar en el presente documento norma H.264 o especificación H.264, o norma o especificación H.264/AVC. El Equipo Mixto de Vídeo (JVT) continúa trabajando en las ampliaciones de H.264/AVC, y desarrollando nuevas normas, por ejemplo, para HEVC.

Tanto el codificador de vídeo 20 como el descodificador de vídeo 30 se pueden implementar como cualquiera de una variedad de circuitos de codificador adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), matrices de puertas programables por campo (FPGA), lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. Tanto el codificador de vídeo 20 como el descodificador de vídeo 30 se pueden incluir en uno o más codificadores o descodificadores, cualquiera de los cuales se puede integrar como parte de un codificador/descodificador (CÓDEC) combinado en una respectiva cámara, ordenador, dispositivo móvil, dispositivo de abonado, dispositivo de radiodifusión, caja de conexión, servidor o similares.

Una secuencia de vídeo incluye típicamente una serie de tramas de vídeo. Un grupo de imágenes (GOP) comprende en general una serie de una o más tramas de vídeo. Un GOP puede incluir datos de sintaxis en una cabecera del GOP, una cabecera de una o más tramas del GOP, o en otras ubicaciones, que describen el número de tramas incluidas en el GOP. Cada trama puede incluir datos de sintaxis de trama que describen un modo de codificación para la respectiva trama. El codificador de vídeo 20 funciona típicamente sobre bloques de vídeo dentro de tramas de vídeo individuales para codificar los datos de vídeo. Un bloque de vídeo puede corresponder a un macrobloque o una partición de un macrobloque. Los bloques de vídeo pueden tener tamaños fijos o variables y pueden diferir en tamaño de acuerdo con una norma de codificación especificada. Cada trama de vídeo puede incluir una pluralidad de fragmentos. Cada fragmento puede incluir una pluralidad de macrobloques, que se pueden disponer en particiones, también denominadas subbloques.

En un ejemplo, la norma ITU-T H.264 da soporte a la intrapredicción 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 interpredicción en diversos tamaños de bloque, tales como 16x16, 16x8, 8x16, 8x8, 8x4, 4x8 y 4x4 para componentes de luma y

tamaños escalados correspondientes para componentes de croma. En la presente divulgación, "NxN" y "N por N" se pueden usar de forma intercambiable para referirse a las dimensiones de píxel del bloque en términos de dimensiones vertical y horizontal, por ejemplo, 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 ( $x = 16$ ). Asimismo, un bloque de NxN tiene, en general, N píxeles en una dirección vertical y N píxeles en una dirección horizontal, donde N representa un valor entero no negativo. Los píxeles de un bloque se pueden disponer en filas y columnas. Además, no es necesario que los bloques tengan necesariamente el mismo número de píxeles en la dirección horizontal que en la dirección vertical. Por ejemplo, los bloques pueden comprender NxM píxeles, donde M no es necesariamente igual a N.

Los tamaños de bloque que son inferiores a 16 por 16 pueden denominarse particiones de un macrobloque de 16 por 16. Los bloques de vídeo pueden comprender bloques de datos de píxeles en el dominio de píxel, o bloques de coeficientes de transformada en el dominio de transformada, por ejemplo, después de la aplicación de una transformada, tal como una transformada discreta del coseno (DCT), una transformada de números enteros, una transformada de ondículas o una transformada conceptualmente similar a los datos de bloques de vídeo residuales que representan diferencias de píxeles entre los bloques de vídeo codificados y los bloques de vídeo predictivos. En algunos casos, un bloque de vídeo puede comprender bloques de coeficientes de transformada cuantificados en el dominio de transformada.

Los bloques de vídeo más pequeños pueden proporcionar una mejor resolución y se pueden usar para localizaciones de una trama de vídeo que incluyen altos niveles de detalle. En general, los macrobloques y las diversas particiones, denominadas en ocasiones subbloques, se pueden considerar bloques de vídeo. Además, se puede considerar que un fragmento es una pluralidad de bloques de vídeo, tales como macrobloques y/o subbloques. Cada fragmento puede ser una unidad descodificable independientemente de una trama de vídeo. De forma alternativa, las propias tramas pueden ser unidades descodificables, o se pueden definir otras porciones de una trama como unidades descodificables. El término "unidad codificada" se puede referir a cualquier unidad descodificable independientemente de una trama de vídeo, tal como una trama completa, un fragmento de una trama, un grupo de imágenes (GOP), denominado también una secuencia, u otra unidad descodificable independientemente definida de acuerdo con las técnicas de codificación aplicables.

Recientemente se ha finalizado una nueva norma de codificación de vídeo, denominada codificación de vídeo de alta eficiencia (HEVC). Actualmente se están realizando esfuerzos para desarrollar diversas extensiones para HEVC, incluida una extensión denominada la extensión de la codificación de contenido de pantalla. Los trabajos de normalización de HEVC se basan en un modelo de dispositivo de codificación de vídeo de modelo de prueba de HEVC (HM). El HM asume varias capacidades de los dispositivos de codificación de vídeo respecto a los dispositivos de acuerdo con, por ejemplo, la norma UIT-T H.264/AVC. Por ejemplo, mientras que la norma H.264 proporciona nueve modos de codificación mediante intrapredicción, el HM proporciona hasta treinta y cuatro modos de codificación mediante intrapredicción.

El HM se refiere a un bloque de datos de vídeo como una unidad de codificación (CU). Los datos de sintaxis dentro de un flujo de bits pueden definir una unidad de codificación más grande (LCU), que es una unidad de codificación más grande en términos del número de píxeles. En general, una CU tiene un propósito similar a un macrobloque de H.264, excepto en que una CU no tiene distinción de tamaño. Por tanto, una CU se puede dividir en sub-CU. En general, las referencias en la presente divulgación a una CU se pueden referir a una unidad de codificación más grande que una imagen o a una sub-CU de una LCU. Una LCU se puede dividir en sub-CU, y cada sub-CU se puede dividir en sub-CU. Los datos de sintaxis para un flujo de bits pueden definir un número máximo de veces que se puede separar una LCU, denominado profundidad de CU. En consecuencia, un flujo de bits también puede definir la unidad de codificación más pequeña (SCU). Esta divulgación también usa el término "bloque" para referirse a una CU, una PU o una TU. Además, cuando esta divulgación se refiere a ejemplos que implican una unidad de codificación o CU, debe entenderse que pueden proporcionarse otros ejemplos con respecto a macrobloques sustituidos por unidades de codificación.

Una LCU se puede asociar con una estructura de datos de árbol cuaternario. En general, una estructura de datos de árbol cuaternario 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, de los que cada uno corresponde a una de las sub-CU. Cada nodo de la estructura de datos de árbol cuaternario puede proporcionar datos de sintaxis para la CU correspondiente. Por ejemplo, un nodo en el árbol cuaternario puede incluir un indicador de división, que indica si la CU correspondiente al nodo está dividida en diversas sub-CU. Los elementos de sintaxis para una CU se pueden definir de forma recursiva y pueden depender de si la CU está dividida en diversas sub-CU.

Una CU que no está dividida (por ejemplo, que corresponde a un nodo hoja en la estructura de datos de árbol cuaternario) puede incluir una o más unidades de predicción (PU). En general, una PU representa la totalidad o una porción de la CU correspondiente, e incluye datos para recuperar una muestra de referencia para la PU. Por ejemplo, cuando la PU se codifica mediante intramodo, la PU puede incluir datos que describen un modo de intrapredicción para la PU. Como otro ejemplo, cuando la PU se codifica en intermodo, 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 (por ejemplo, una precisión de un píxel entero, una precisión de un cuarto de píxel o una precisión de un octavo de píxel), una trama de referencia a la que apunta el vector de movimiento y/o una lista de referencia (por ejemplo, lista 0 o lista 1) para el vector de movimiento. Los datos para la CU que definen la(s) PU también pueden describir, por ejemplo, una partición de la CU en una o más PU. Los modos de partición pueden diferir entre si la CU está sin codificar, codificada en modo de intrapredicción o codificada en modo de interpredicción.

Una CU que tiene una o más PU también puede incluir una o más unidades de transformada (TU). Después de la predicción usando una PU, un codificador de vídeo puede calcular un valor residual para la porción de la CU correspondiente a la PU. El valor residual puede transformarse, cuantificarse y escanearse. Una TU no se limita necesariamente al tamaño de una PU. Por tanto, las TU pueden ser más grandes o más pequeñas que las PU correspondientes para la misma CU. En algunos ejemplos, el tamaño máximo de una TU puede corresponder al tamaño de la CU que incluye la TU.

De acuerdo con las técnicas de esta divulgación, el codificador de vídeo 20 puede codificar en intermodo una CU que utiliza una o más PU que tienen vectores de movimiento con una precisión de un píxel subentero variable. Por ejemplo, el codificador de vídeo 20 puede seleccionar entre usar un vector de movimiento que tenga una precisión de un píxel entero o una precisión de píxel fraccionaria (por ejemplo, un cuarto o un octavo) para una PU basándose en el contenido de los datos de vídeo que se codifican. De acuerdo con algunas técnicas de esta divulgación, el codificador de vídeo 20 puede no necesitar generar, para su inclusión en el flujo de bits de datos de vídeo codificados, una indicación de la precisión de subpíxel para un vector de movimiento de una PU. En su lugar, el descodificador de vídeo 30 puede derivar la precisión del vector de movimiento usando las mismas técnicas de derivación utilizadas por el codificador de vídeo 20. De acuerdo con otras técnicas de esta divulgación, el codificador de vídeo 20 puede incluir, en el flujo de bits de los datos de vídeo codificados, uno o más elementos de sintaxis que el descodificador de vídeo 30 puede utilizar para determinar la precisión del vector de movimiento seleccionada.

Para calcular valores para las posiciones de píxeles subenteros, el codificador de vídeo 20 puede incluir una variedad de filtros de interpolación. Por ejemplo, la interpolación bilineal puede usarse para calcular los valores de las posiciones de píxeles subenteros. El codificador de vídeo 20 puede configurarse para realizar una búsqueda del movimiento con respecto a los datos de luminancia de una PU para calcular un vector de movimiento que utiliza los datos de luminancia de la PU. El codificador de vídeo 20 puede entonces reutilizar el vector de movimiento para codificar los datos de crominancia de la PU. Típicamente, los datos de crominancia tienen una resolución más baja que los datos de luminancia correspondientes, por ejemplo, un cuarto de la resolución de los datos de luminancia. Por lo tanto, el vector de movimiento para los datos de crominancia puede tener una precisión mayor que para los datos de luminancia. Por ejemplo, los vectores de movimiento con precisión de un cuarto de píxel para los datos de luminancia pueden tener una precisión de un octavo de píxel para los datos de crominancia. De manera similar, los vectores de movimiento con precisión de un octavo de píxel para los datos de luminancia pueden tener una precisión de un dieciseisavo de píxel para los datos de crominancia.

Después de una codificación intrapredictiva o interpredictiva para generar datos predictivos y datos residuales, y después de cualquier transformada (tal como la transformada de números enteros de 4x4 o de 8x8 usada en la norma H.264/AVC o una transformada discreta del coseno DCT) para producir coeficientes de transformada, se puede realizar la cuantificación de los coeficientes de transformada. La cuantificación se refiere, en general, a un proceso en el que los coeficientes de transformada se cuantifican para reducir, posiblemente, la cantidad de datos que se utilizan para representar los coeficientes. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos, o la totalidad, de los coeficientes. Por ejemplo, un valor de n bits se puede redondear a la baja hasta un valor de m bits durante la cuantificación, donde n es mayor que m.

Después de la cuantificación, se puede realizar la codificación por entropía de los datos cuantificados, por ejemplo, de acuerdo con la codificación de longitud variable adaptable según el contenido (CAVLC), la codificación aritmética binaria adaptable 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 de ejecución cero de coeficientes cuantificados y/o la generación de información de sintaxis, tal como valores de patrón de bloque codificado (CBP), tipo macrobloque, modo de codificación, tamaño de LCU o similares.

El descodificador de vídeo 30 del dispositivo de destino 14 puede configurarse para realizar técnicas similares y, en general, simétricas, a cualquiera o a todas las técnicas del codificador de vídeo 20 de esta divulgación. Por ejemplo, el descodificador de vídeo 30 puede configurarse para recibir información que define un contexto en el que se ha codificado una indicación de una precisión de subpíxeles para un vector de movimiento de una PU de una CU. El codificador de vídeo 20 puede proporcionar, y el descodificador de vídeo 30 puede recibir, la información de contexto en un árbol cuaternario para una LCU que incluye la CU y la PU. La información de contexto puede corresponder a información de tamaño para la CU y/o la PU, por ejemplo, una profundidad de la CU, un tamaño de la PU y/o un tipo de la PU. El descodificador de vídeo 30 puede usar la información de contexto para descodificar la indicación de la precisión de subpíxel del vector de movimiento, por ejemplo, para determinar si el vector de movimiento tiene una precisión de un cuarto de píxel o una precisión de un octavo de píxel. Por ejemplo, el descodificador de vídeo 30 puede

realizar un proceso de codificación por entropía inversa usando la información de contexto para descodificar por entropía la indicación de la precisión de subpíxel para el vector de movimiento.

Tanto el codificador de vídeo 20 como el descodificador de vídeo 30 se pueden implementar como cualquiera de una variedad de circuitos codificadores o descodificadores adecuados, según sea aplicable, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), matrices de puertas programables por campo (FPGA), circuitos de lógica discreta, programas informáticos, hardware, firmware o cualquier combinación de los mismos. Tanto el codificador de vídeo 20 como el descodificador de vídeo 30 se pueden incluir en uno o más codificadores o descodificadores, ambos de los cuales se pueden integrar como parte de un codificador/descodificador (CÓDEC) de vídeo combinado. Un aparato que incluye un codificador de vídeo 20 y/o un descodificador de vídeo 30 puede comprender un circuito integrado, un microprocesador y/o un dispositivo de comunicación inalámbrica, tal como un teléfono celular.

La FIG. 2 es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo 20 que puede implementar técnicas para dar soporte a la resolución adaptable del vector de movimiento. El codificador de vídeo 20 puede realizar la intrapredicción e interpredicción de bloques dentro de tramas de vídeo, que incluyen LCU, CU y PU, y calcular valores residuales que pueden codificarse como TU. La intracodificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial en el vídeo dentro de una trama de vídeo dada. La intercodificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal en el vídeo dentro de tramas contiguas de una secuencia de vídeo. El intramodo (modo I) puede referirse a cualquiera de varios modos de compresión con base espacial, y los intermodos tales como la predicción unidireccional (modo P) o la predicción bidireccional (modo B) pueden referirse a cualquiera de varios modos de compresión con base temporal. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden realizar la codificación mediante interpredicción, mientras que la unidad de intrapredicción 46 puede realizar la codificación mediante intrapredicción.

Como se muestra en la FIG. 2, el codificador de vídeo 20 recibe un bloque de vídeo actual dentro de una trama de vídeo que se va a codificar. En el ejemplo de la FIG. 2, el codificador de vídeo 20 incluye una memoria de datos de vídeo 38, una unidad de compensación de movimiento 44, una unidad de estimación de movimiento 42, una unidad de intrapredicción 46, un búfer de imágenes descodificadas 64, un sumador 50, una unidad de transformada 52, una unidad de cuantificación 54 y una unidad de codificación por entropía 56. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 incluye también una unidad de cuantificación inversa 58, una unidad de transformada inversa 60 y un sumador 62. También se puede incluir un filtro de eliminación de bloques (no se muestra en la FIG. 2) que filtra los límites de bloque, para eliminar distorsiones de efecto pixelado del vídeo reconstruido. Si se desea, el filtro de eliminación de bloques filtrará típicamente la salida del sumador 62.

La memoria de datos de vídeo 38 puede almacenar datos de vídeo que se van a codificar mediante los componentes del codificador de vídeo 20. Los datos de vídeo almacenados en la memoria de datos de vídeo 38 se pueden obtener, por ejemplo, a partir de la fuente de vídeo 18. El búfer de imágenes descodificadas 64 puede ser una memoria de imágenes de referencia que almacena datos de vídeo de referencia para su uso en la codificación de datos de vídeo mediante el codificador de vídeo 20, por ejemplo, en los modos de intracodificación o intercodificación. La memoria de datos de vídeo 38 y el búfer de imágenes descodificadas 64 pueden estar formadas por cualquiera de una variedad de dispositivos de memoria, tales como la memoria dinámica de acceso aleatorio (DRAM), que incluye la DRAM síncrona (SDRAM), la RAM magnetorresistiva (MRAM), la RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. El mismo dispositivo de memoria u otros dispositivos de memoria separados pueden proporcionar una memoria de datos de vídeo 38 y un búfer de imágenes descodificadas 64. En diversos ejemplos, la memoria de datos de vídeo 38 puede estar en un chip con otros componentes del codificador de vídeo 20, o fuera del chip en relación con esos componentes.

Durante el proceso de codificación, el codificador de vídeo 20 recibe una trama o un fragmento de vídeo que se va a codificar. La trama o el fragmento se puede dividir en múltiples bloques de vídeo (por ejemplo, LCU). La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 realizan una codificación interpredictiva del bloque de vídeo recibido en relación con uno o más bloques de una o más tramas de referencia para proporcionar una compresión temporal. La unidad de intrapredicción 46 puede llevar a cabo una codificación intrapredictiva del bloque de vídeo recibido en relación con uno o más bloques contiguos de la misma trama o fragmento que el bloque que se va a codificar para proporcionar compresión espacial.

La unidad de selección de modo 40 puede seleccionar uno de los modos de codificación, intra o inter, por ejemplo, basándose en los resultados de errores, y proporciona el bloque intracodificado o intercodificado 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. Cuando la unidad de selección de modo 40 selecciona la codificación en intermodo para un bloque, la unidad de selección de resolución 48 puede seleccionar una resolución para un vector de movimiento del bloque. Por ejemplo, la unidad de selección de resolución 48 puede seleccionar una precisión de un octavo de píxel o una precisión de un cuarto de píxel para un vector de movimiento del bloque.

Como ejemplo, la unidad de selección de resolución 48 puede configurarse para comparar una diferencia de error entre usar un vector de movimiento con una precisión de un cuarto de píxel para codificar un bloque y usar un vector



de movimiento con una precisión de un octavo de píxel para codificar el bloque. La unidad de estimación de movimiento 42 puede configurarse para codificar un bloque usando uno o más vectores de movimiento con precisión de un cuarto de píxel en una primera pasada de codificación y uno o más vectores de movimiento con precisión de octavo de píxel en una segunda pasada de codificación. La unidad de estimación de movimiento 42 puede usar además una variedad de combinaciones de uno o más vectores de movimiento con precisión de un cuarto de píxel y uno o más vectores de movimiento con precisión de un octavo de píxel para el bloque en una tercera pasada de codificación. La unidad de selección de resolución 48 puede calcular valores entre distorsión y velocidad para cada pasada de codificación del bloque y calcular las diferencias entre los valores de distorsión y velocidad.

Cuando la diferencia sobrepasa un umbral, la unidad de selección de resolución 48 puede seleccionar el vector de movimiento con precisión de un octavo de píxel para codificar el bloque. La unidad de selección de resolución 48 también puede evaluar la información entre distorsión y velocidad, analizar un presupuesto de bits y/o analizar otros factores para determinar si se debe usar una precisión de un octavo de píxel o una precisión de un cuarto de píxel para un vector de movimiento al codificar un bloque durante un proceso de predicción en intermodo. Después de seleccionar una precisión de un octavo de píxel o una precisión de un cuarto de píxel para que un bloque sea codificado en intermodo, la unidad de selección de modo 40 o la estimación de movimiento pueden enviar un mensaje (por ejemplo, una señal) a la unidad de estimación de movimiento 42 indicativo de la precisión seleccionada para un vector de movimiento.

La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden estar altamente integradas, pero se ilustran por separado con propósitos conceptuales. La estimación de movimiento es el proceso de generación de vectores de movimiento, que estiman el movimiento para bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de un bloque predictivo de una trama de referencia predictiva (u otra unidad codificada) en relación con el bloque actual que se está codificando dentro de la trama actual (u otra unidad codificada). Un bloque predictivo es un bloque que se considera estrechamente coincidente con el bloque que se va a codificar, en términos de diferencia de píxeles, lo cual se puede determinar mediante una suma de diferencias absolutas (SAD), una suma de diferencias al cuadrado (SSD) u otras métricas de diferencia. Un vector de movimiento también puede indicar el desplazamiento de una partición de un macrobloque. La compensación de movimiento puede implicar obtener o generar el bloque predictivo en base al vector de movimiento determinado por la estimación de movimiento. De nuevo, la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 se pueden integrar funcionalmente, en algunos ejemplos.

La unidad de estimación de movimiento 42 calcula un vector de movimiento para el bloque de vídeo de una trama intercodificada comparando el bloque de vídeo con los bloques de vídeo de una trama de referencia en el búfer de imágenes descodificadas 64. La unidad de compensación de movimiento 44 también puede interpolar píxeles subenteros de la trama de referencia, por ejemplo, una trama I o una trama P. La norma UIT H.264, como ejemplo, describe dos listas: la lista 0, que incluye tramas de referencia que tienen un orden de visualización anterior a una trama actual que se está codificando, y la lista 1, que incluye tramas de referencia que tienen un orden de visualización posterior a la trama actual que se está codificando. Por lo tanto, los datos almacenados en el búfer de imágenes descodificadas 64 se pueden organizar de acuerdo con estas listas.

De acuerdo con las técnicas de esta divulgación, la unidad de compensación de movimiento 44 puede configurarse para interpolar valores en posiciones de un dieciseisavo de píxel de datos de crominancia de una CU cuando un vector de movimiento para los datos de luminancia de la CU tiene una precisión de un octavo de píxel. Para interpolar valores en las posiciones de un dieciseisavo de píxel de los datos de crominancia, la unidad de compensación de movimiento 44 puede utilizar la interpolación bilineal. Por lo tanto, el sumador 50 puede calcular un residuo para los datos de crominancia de la CU en relación con los valores interpolados bilinealmente de las posiciones de un dieciseisavo de píxel de un bloque de referencia. De esta manera, el codificador de vídeo 20 puede calcular, usando interpolación bilineal, valores de posiciones de un dieciseisavo de píxel de los datos de crominancia de un bloque de referencia identificado por un vector de movimiento y codificar los datos de crominancia de una unidad de codificación en base a los valores interpolados bilinealmente del bloque de referencia, cuando los datos de luminancia de la unidad de codificación se han codificado usando un vector de movimiento que tiene una precisión de un octavo de píxel para los datos de luminancia.

La unidad de estimación de movimiento 42 compara bloques de una o más tramas de referencia del búfer de imágenes descodificadas 64 con un bloque a codificar de una trama actual, por ejemplo, una trama P o una trama B. Cuando las tramas de referencia en el búfer de imágenes descodificadas 64 incluyen valores para píxeles subenteros, un vector de movimiento calculado mediante la unidad de estimación de movimiento 42 puede referirse a una ubicación de píxel subentero de una trama de referencia. La unidad de estimación de movimiento 42 y/o la unidad de compensación de movimiento 44 también puede configurarse para calcular los valores de las posiciones de píxeles subenteros de tramas de referencia almacenadas en el búfer de imágenes descodificadas 64 si no hay valores de posiciones de píxeles subenteros almacenados en el búfer de imágenes descodificadas 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 se puede denominar bloque predictivo.

La unidad de estimación de movimiento 42, la unidad de compensación de movimiento 44, la unidad de selección de modo 40 u otra unidad del codificador de vídeo 20, también pueden indicar el uso de una precisión de un cuarto de píxel o una precisión de un octavo de píxel para un vector de movimiento utilizado para codificar un bloque. Por ejemplo, la unidad de estimación de movimiento 42 puede enviar una indicación de una precisión de píxel subentero para el vector de movimiento a la unidad de codificación por entropía 56. La unidad de estimación de movimiento 42 también puede proporcionar información de contexto relacionada con la información de tamaño para una PU correspondiente al vector de movimiento a la unidad de codificación por entropía 56, donde la información de tamaño puede incluir parte o toda la profundidad de una CU que incluye la PU, un tamaño de la PU y/o un tipo de la PU.

La unidad de compensación de movimiento 44 puede calcular los datos de predicción en base al bloque predictivo. El codificador de vídeo 20 forma un bloque de vídeo residual restando los datos de predicción desde la unidad de compensación de movimiento 44 del bloque de vídeo original que se esté codificando. El sumador 50 representa el componente o los componentes que realizan esta operación de resta. La unidad de transformada 52 aplica una transformada, tal como una transformada discreta del coseno (DCT) o una transformada conceptualmente similar, al bloque residual, produciendo un bloque de vídeo que comprende valores de coeficientes de transformada residuales.

La unidad de transformada 52 puede realizar otras transformadas, tales como las definidas por la norma H.264, que son conceptualmente similares a la DCT. También se podrían usar transformadas de ondículas, transformadas de números enteros, transformadas de subbanda u otros tipos de transformadas. En cualquier caso, la unidad de transformada 52 aplica la transformada al bloque residual, produciendo un bloque de coeficientes de transformada residuales. La transformada puede convertir la información residual desde un dominio de valores de píxel a un dominio de transformada, tal como un dominio de la frecuencia. La unidad de cuantificación 54 cuantifica los coeficientes de transformada residuales para reducir más la velocidad de transmisión de bits. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos, o la totalidad, de los coeficientes. El grado de cuantificación se puede modificar ajustando un parámetro de cuantificación.

Después de la cuantificación, la unidad de codificación por entropía 56 codifica por entropía los coeficientes de transformada cuantificados. Por ejemplo, la unidad de codificación por entropía 56 puede realizar contenido CAVLC, CABAC u otra técnica de codificación por entropía. Después de la codificación por entropía realizada por la unidad de codificación por entropía 56, el vídeo codificado se puede transmitir a otro dispositivo o archivarse para su transmisión o recuperación posterior. En el caso de una codificación aritmética binaria adaptable según el contexto, el contexto puede basarse en macrobloques contiguos.

En algunos casos, la unidad de codificación por entropía 56 u otra unidad de codificador de vídeo 20 pueden estar configuradas para realizar otras funciones de codificación, además de la codificación por entropía. Por ejemplo, la unidad de codificación por entropía 56 puede estar configurada para determinar los valores CBP de los macrobloques y las particiones. También, en algunos casos, la unidad de codificación por entropía 56 puede realizar una codificación de longitud de ejecución de los coeficientes en un macrobloque o partición del mismo. En particular, la unidad de codificación por entropía 56 puede aplicar un escaneado en zigzag u otro patrón de escaneado para escanear los coeficientes de transformada en un macrobloque o partición y codificar secuencias de ceros para una mayor compresión. La unidad de codificación por entropía 56 también puede construir la información de cabecera con los elementos de sintaxis adecuados para la transmisión en el flujo de bits de vídeo codificado.

De acuerdo con las técnicas de esta divulgación, en los casos en que la precisión de subpíxel se señala en lugar de derivarse, la unidad de codificación por entropía 56 puede configurarse para codificar una indicación de precisión de subpíxel para un vector de movimiento, por ejemplo, para indicar si el vector de movimiento tiene una precisión de un píxel entero o una precisión de subpíxel, tal como una precisión de un cuarto de píxel o una precisión de un octavo de píxel (u otras precisiones de subpíxel, en diversos ejemplos). La unidad de codificación por entropía 56 puede codificar la indicación utilizando CABAC. Además, la unidad de codificación por entropía 56 puede usar información de contexto para realizar CABAC para codificar la indicación que indica la información de tamaño para una PU correspondiente al vector de movimiento, donde la información de tamaño puede incluir parte o toda la profundidad de una CU que incluye la PU, un tamaño de la PU y/o un tipo de la PU.

El codificador de vídeo 20 puede señalar de forma predictiva el vector de movimiento. Dos ejemplos de técnicas de señalización predictiva que pueden implementarse mediante el codificador de vídeo 20 incluyen la predicción avanzada del vector de movimiento (AMVP) y la señalización del modo de fusión. En AMVP, el codificador de vídeo 20 y el decodificador de vídeo 30 reúnen listas de candidatos basándose en vectores de movimiento determinados a partir de bloques ya codificados. A continuación, el codificador de vídeo 20 señala un índice en la lista de candidatos para identificar un predictor del vector de movimiento (MVP) y señala una diferencia del vector de movimiento (MVD). El decodificador de vídeo 30 interpredice un bloque usando el MVP modificado por la MVD, por ejemplo, usando un vector de movimiento igual a  $MVP + MVD$ .

En el modo de fusión, el codificador de vídeo 20 y el decodificador de vídeo 30 ambos reúnen una lista de candidatos basada en bloques ya codificados, y el codificador de vídeo 20 señala un índice para uno de los candidatos en la lista de candidatos. En el modo de fusión, el decodificador de vídeo 30 interpredice el bloque actual utilizando el vector de movimiento y el índice de imagen de referencia del candidato señalado. Tanto en el modo AMVP como en

el modo de fusión, el codificador de vídeo 20 y el descodificador de vídeo 30 utilizan las mismas técnicas de construcción de listas, de modo que la lista utilizada por el codificador de vídeo 20 al determinar cómo codificar un bloque coincide con la lista utilizada por el descodificador de vídeo 30 al determinar cómo descodificar el bloque.

La unidad de cuantificación inversa 58 y la unidad de transformada inversa 60 aplican la cuantificación inversa y la transformación inversa, respectivamente, para reconstruir el bloque residual en el dominio de píxel, por ejemplo, para su uso posterior como bloque de referencia. La unidad de compensación de movimiento 44 puede calcular un bloque de referencia añadiendo el bloque residual a un bloque predictivo de una de las tramas del búfer de imágenes descodificadas 64. La unidad de compensación de movimiento 44 también puede aplicar uno o más filtros de interpolación al bloque residual reconstruido para calcular los valores de píxel subentero, para su uso en la estimación de movimiento. El sumador 62 añade el bloque residual reconstruido al bloque de predicción compensado por movimiento, producido por la unidad de compensación de movimiento 44 para producir un bloque de vídeo reconstruido para su almacenamiento en el búfer de imágenes descodificadas 64. El bloque de vídeo reconstruido se puede usar por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 como un bloque de referencia para intercodificar un bloque en una trama de vídeo posterior.

La FIG. 3 es un diagrama de bloques que ilustra un descodificador de vídeo de ejemplo 30, que descodifica una secuencia de vídeo codificada. 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 intrapredicción 74, una unidad de cuantificación inversa 76, una unidad de transformación inversa 78, un búfer de imágenes descodificadas 82 y un sumador 80. En algunos ejemplos, el descodificador de vídeo 30 puede realizar una pasada de descodificación, en general, recíproca a la pasada de codificación descrita con respecto al codificador de vídeo 20 (FIG. 2). La unidad de compensación de movimiento 72 puede generar datos de predicción en base a vectores de movimiento recibidos desde la unidad de descodificación por entropía 70.

La memoria de datos de vídeo 68 puede almacenar los datos de vídeo, tales como un flujo de bits de vídeo codificado, que se van a descodificar mediante los componentes del descodificador de vídeo 30. Los datos de vídeo almacenados en la memoria de datos de vídeo 68 pueden obtenerse, por ejemplo, a partir de un medio legible por ordenador 16, por ejemplo, desde una fuente de vídeo local, tal como una cámara, mediante comunicación de datos de vídeo por red alámbrica o inalámbrica, o accediendo a medios de almacenamiento físico de datos. La memoria de datos de vídeo 68 puede formar un búfer de imágenes codificadas (CPB) que almacena datos de vídeo codificados a partir de un flujo de bits de vídeo codificado. El búfer de imágenes descodificadas 82 puede ser una memoria de imágenes de referencia que almacena datos de vídeo de referencia para su uso en la descodificación de datos de vídeo mediante el descodificador de vídeo 30, por ejemplo, en los modos de intracodificación o intercodificación. La memoria de datos de vídeo 68 y el búfer de imágenes descodificadas 82 pueden estar formadas por cualquiera de una variedad de dispositivos de memoria, tales como la memoria dinámica de acceso aleatorio (DRAM), incluyendo la DRAM síncrona (SDRAM), la RAM magnetorresistiva (MRAM), la RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. El mismo dispositivo de memoria u otros dispositivos de memoria separados pueden proporcionar una memoria de datos de vídeo 68 y un búfer de imágenes descodificadas 82. En diversos ejemplos, la memoria de datos de vídeo 68 puede estar en un chip con otros componentes del descodificador de vídeo 30, o fuera del chip en relación con esos componentes.

La unidad de descodificación por entropía 70 puede recuperar un flujo de bits codificado, por ejemplo, de la memoria de datos de vídeo 68. El flujo de bits codificado puede incluir datos de vídeo codificados por entropía. La unidad de descodificación por entropía 70 puede descodificar los datos de vídeo codificados por entropía, y a partir de los datos de vídeo descodificados por entropía, la unidad de compensación de movimiento 72 puede determinar información de movimiento, que incluye vectores de movimiento, precisión del vector de movimiento, índices de listas de imágenes de referencia y otra información de movimiento. La unidad de compensación de movimiento 72 puede, por ejemplo, determinar dicha información realizando las técnicas de AMVP y de modo de fusión descritos anteriormente.

La unidad de compensación de movimiento 72 puede usar los vectores de movimiento recibidos en el flujo de bits para identificar un bloque de predicción en las tramas de referencia del búfer de imágenes descodificadas 82. La unidad de intrapredicción 74 puede usar modos de intrapredicción recibidos en el flujo de bits para formar un bloque de predicción a partir de bloques espacialmente contiguos. La unidad de cuantificación inversa 76 realiza la cuantificación inversa, es decir, descuantifica, los coeficientes de bloque cuantificados proporcionados en el flujo de bits y descodificados mediante la unidad de descodificación por entropía 70. El proceso de cuantificación inversa puede incluir un proceso convencional, por ejemplo, como se define en la norma de descodificación H.264. El proceso de cuantificación inversa también puede incluir el uso de un parámetro de cuantificación QP calculado mediante el codificador de vídeo 20 para cada macrobloque con el fin de determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa que debería aplicarse.

La unidad de transformada inversa 58 aplica una transformada inversa, por ejemplo, una DCT inversa, una transformada inversa de números enteros, o un proceso de transformada inversa conceptualmente similar a los coeficientes de transformada para producir bloques residuales en el dominio de píxel. La unidad de compensación de movimiento 72 produce bloques con compensación de movimiento, posiblemente realizando una interpolación basada en filtros de interpolación. Los identificadores para filtros de interpolación que se van a usar para la estimación de

movimiento con una precisión de subpíxel se pueden incluir en los elementos de sintaxis. La unidad de compensación de movimiento 72 puede usar filtros de interpolación como se usan por el codificador de vídeo 20 durante la codificación del bloque de vídeo, para calcular valores interpolados para píxeles subenteros 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 con la información de sintaxis recibida y usar los filtros de interpolación para producir bloques predictivos. De acuerdo con las técnicas de esta divulgación, la unidad de compensación de movimiento 72 puede interpolar valores de posiciones de un dieciseisavo de píxel de los datos de crominancia de un bloque de referencia cuando un vector de movimiento tiene una precisión de un octavo de píxel para los datos de luminancia. Por ejemplo, la unidad de compensación de movimiento 72 puede usar interpolación bilineal para interpolar los valores de las posiciones de un dieciseisavo de píxel del bloque de referencia.

La unidad de compensación de movimiento 72 utiliza parte de la información de sintaxis para determinar tamaños de LCU y CU que se utilizan para codificar trama(s) y/o fragmento(s) de la secuencia de vídeo codificada, información de partición que describe cómo cada macrobloque de una trama de la secuencia de vídeo codificada está dividida, modos que indican cómo cada partición está codificada, una o más tramas de referencia (y listas de tramas de referencia) para cada CU intercodificada y otra información para descodificar la secuencia de vídeo codificada.

El sumador 80 suma los bloques residuales a los bloques de predicción correspondientes generados por la unidad de compensación de movimiento 72 o por la unidad de intrapredicción para formar bloques descodificados. Si se desea, también se puede aplicar un filtro de eliminación de bloques para filtrar los bloques descodificados para eliminar distorsiones de efecto pixelado. Los bloques de vídeo descodificado se almacenan a continuación en el búfer de imágenes descodificadas 82, que proporciona bloques de referencia para una posterior compensación de movimiento y también proporciona vídeo descodificado para su presentación en un dispositivo de visualización (tal como el dispositivo de visualización 32 de la FIG. 1).

La FIG. 4 es un diagrama conceptual que ilustra posiciones de píxeles fraccionarios para una posición de píxel completo. En particular, la FIG. 4 ilustra posiciones de píxeles fraccionarios para un píxel completo (pel) 100. El píxel completo 100 corresponde a las posiciones de medio píxel 102A-102C (medio pel 102), las posiciones de un cuarto de píxel 104A-104L (cuarto de pel 104) y las posiciones de un octavo de píxel 106A-106AV (octavo de pel 106).

La FIG. 4 ilustra las posiciones de octavo de píxel 106 de un bloque usando un contorno en línea discontinua para indicar que estas posiciones pueden incluirse opcionalmente. Es decir, si un vector de movimiento tiene una precisión de un octavo de píxel, el vector de movimiento puede apuntar a cualquier posición de píxel completo 100, a posiciones de medio píxel 102, a posiciones de un cuarto de píxel 104 o a posiciones de octavo de píxel 106. Sin embargo, si el vector de movimiento tiene una precisión de un cuarto de píxel, el vector de movimiento puede apuntar a cualquier posición de píxel completo 100, a posiciones de medio píxel 102 o a posiciones de un cuarto de píxel 104, pero no apuntará a posiciones de octavo de píxel 106. Debe entenderse además que en otros ejemplos, se pueden utilizar otras precisiones, por ejemplo, precisión de un dieciseisavo de píxel, precisión de un treintadosavo de píxel, o similares.

Puede incluirse un valor para el píxel en la posición de píxel completo 100 en una trama de referencia correspondiente. Es decir, el valor del píxel en la posición de píxel completo 100 en general corresponde al valor real de un píxel en la trama de referencia, por ejemplo, que en última instancia se representa y se visualiza cuando se visualiza la trama de referencia. Los valores para las posiciones de medio píxel 102, las posiciones de un cuarto de píxel 104 y las posiciones de octavo píxel 106 (denominadas colectivamente posiciones de píxel fraccionario) pueden interpolarse utilizando filtros de interpolación adaptables o filtros de interpolación fijos, por ejemplo, filtros de diversos números de "taps" (coeficientes) como diversos filtros de Wiener, filtros bilineales u otros filtros. En general, el valor de una posición de píxel fraccionario puede interpolarse a partir de uno o más píxeles contiguos, que corresponden a valores de posiciones contiguas de píxel completo o posiciones de píxeles fraccionarios determinadas previamente.

De acuerdo con las técnicas de esta divulgación, un codificador de vídeo, tal como el codificador de vídeo 20, puede seleccionar de forma adaptable la precisión para un vector de movimiento, por ejemplo, entre la precisión de un píxel entero o la precisión de un píxel fraccionario, tal como la precisión de un octavo de píxel y la precisión de un cuarto de píxel. El codificador de vídeo 20 puede realizar esta selección para cada vector de movimiento, cada CU, cada LCU, cada fragmento, cada trama, cada GOP u otras unidades codificadas de datos de vídeo. Cuando el codificador de vídeo 20 selecciona una precisión de un cuarto de píxel para un vector de movimiento, el vector de movimiento puede referirse a cualquier posición de píxel completo 100, a posiciones de medio píxel 102 o a posiciones de un cuarto de píxel 104. Cuando el codificador de vídeo 20 selecciona una precisión de un octavo de píxel para un vector de movimiento, el vector de movimiento puede referirse a cualquier posición de píxel completo 100, a posiciones de medio píxel 102, a posiciones de un cuarto de píxel 104 o a posiciones de octavo de píxel 106.

Las FIG. 5A-5C son diagramas conceptuales que ilustran las correspondientes posiciones de píxeles de crominancia y luminancia. Las FIG. 5A-5C también ilustran cómo los vectores de movimiento calculados para los datos de luminancia pueden reutilizarse para los datos de crominancia. Como cuestión preliminar, las FIG. 5A-5C ilustran una fila parcial de posiciones de píxeles. Debe entenderse que, en la práctica, una posición de píxel completo puede tener una cuadrícula rectangular de posiciones de píxeles fraccionarios asociadas, como la que se ilustra en la FIG. 4. El

ejemplo de las FIG. 5A-5C pretenden ilustrar los conceptos descritos en esta divulgación, y no pretenden ser una lista exhaustiva de correspondencias entre las posiciones de píxeles fraccionarios de crominancia y las posiciones de píxeles fraccionarios de luminancia.

Las FIG. 5A-5C ilustran las posiciones de los píxeles de un bloque de luminancia, que incluyen la posición de los píxeles completos de luminancia 110, la posición de medio píxel de luminancia 116, la posición de un cuarto de píxel 112 y las posiciones de un octavo de píxel de luminancia 114A, 114B. Las FIG. 5A-5C también ilustran las posiciones de los píxeles correspondientes de un bloque de crominancia, que incluyen la posición de píxel completo de crominancia 120, la posición de un cuarto de píxel de crominancia 122, la posición de un octavo de píxel de crominancia 124 y las posiciones de un dieciseisavo de píxel de crominancia 126A, 126B. En este ejemplo, el píxel completo de crominancia 120 corresponde al píxel completo de luminancia 110. Además, en este ejemplo, el bloque de crominancia se reduce en un factor de dos horizontal y verticalmente en relación con el bloque de luminancia. Por tanto, un píxel de un cuarto de crominancia 122 corresponde a medio píxel de luminancia 116. De manera similar, un octavo de píxel de crominancia 124 corresponde a un cuarto de píxel de luminancia 112, un dieciseisavo de píxel de crominancia 126A corresponde a un octavo de píxel de luminancia 114A, y un dieciseisavo de píxel de crominancia 126B corresponde a un octavo de píxel de luminancia 114B.

En códecs de vídeo avanzados, tal como H.264/AVC, HEVC y, potencialmente, los códecs sucesores de H.264 y HEVC, el coste en bits de señalizar los vectores de movimiento puede aumentar. Para reducir este coste en bits, se puede utilizar la derivación del MV en el lado del descodificador (DMVD). En S. Kamp y M. Wien, "Decoder-side motion vector derivation for block-based video coding", IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, págs. 1732-1745, diciembre de 2012, se propuso el DMVD basándose en una coincidencia de la plantilla en forma de L.

La FIG. 6 es una ilustración de un ejemplo de coincidencia de la plantilla en forma de L para DMVD. En el ejemplo de la FIG. 6, el bloque actual 132 de la imagen 134 actual se interpredice usando la correspondencia de plantillas. La plantilla 136 define una forma que cubre los bloques contiguos ya descodificados del bloque actual 132. Un descodificador de vídeo (por ejemplo, un descodificador de vídeo 30) puede, por ejemplo, comparar primero los valores de los píxeles incluidos en los bloques contiguos ya descodificados cubiertos por la plantilla 136 con los valores de los píxeles incluidos en los bloques contiguos ya descodificados cubiertos por la plantilla colocalizada 138, que cubre los bloques ubicados en una imagen de referencia de las imágenes de referencia 140. El descodificador de vídeo puede entonces mover la plantilla a otras ubicaciones en la imagen de referencia y comparar los valores de los píxeles cubiertos por la plantilla con los valores de los píxeles incluidos en los bloques contiguos ya descodificados cubiertos por la plantilla 136.

En base a estas comparaciones múltiples, el descodificador de vídeo puede determinar la mejor coincidencia, tal como la mejor coincidencia 142 que se muestra en el ejemplo de la FIG. 6. El descodificador de vídeo puede entonces determinar un desplazamiento entre la mejor coincidencia y la plantilla colocalizada. Este desplazamiento (por ejemplo, el desplazamiento 144 en la FIG. 6) corresponde al vector de movimiento utilizado para predecir el bloque actual 132.

Como se ilustra en la FIG. 6, cuando un bloque se codifica en modo DMVD, el descodificador de vídeo 30 busca el MV del bloque, en lugar de señalizarse directamente al descodificador de vídeo 30. El MV que da lugar a la distorsión mínima por coincidencia de plantillas se selecciona como el MV final para el bloque. Para mantener una alta eficacia en la codificación, puede ser necesario un determinado número de coincidencias de plantilla para que el descodificador 30 seleccione un vector de movimiento candidato como el MV para descodificar el bloque actual, lo que puede aumentar la complejidad de la descodificación.

Para reducir la complejidad de la descodificación en DMVD, se propuso un procedimiento de derivación de MV basada en espejo en Y.-J. Chiu, L. Xu, W. Zhang, H. Jiang, "DECODER-SIDE MOTION ESTIMATION AND WIENER FILTER FOR HEVC", VCIP workshop 2013, Malasia, 17-20 de noviembre de 2013.

La FIG. 7 es un diagrama conceptual que ilustra un ejemplo de derivación del MV bidireccional basada en espejos. Como se ilustra en la FIG. 7, la derivación del MV bidireccional basada en espejo se puede aplicar mediante la estimación del movimiento centrosimétrica alrededor de los centros de búsqueda con una exactitud fraccionaria de la muestra en el lado del descodificador. El tamaño/ubicación de la ventana de búsqueda puede estar predefinido y señalizado en el flujo de bits. En la FIG. 7, dMV es una desviación que se suma a PMV0 y se resta de PMV1 para generar un par de MV, MV0 y MV1. Todos los valores de dMV dentro de una ventana de búsqueda se pueden verificar y la suma de la diferencia absoluta (SAD) entre la referencia L0 y los bloques de referencia L1 se puede usar como medición de la estimación del movimiento centrosimétrica. Se puede seleccionar un par de MV con un SAD mínimo como MV finales para el bloque.

Para la resolución del vector de movimiento adaptable, la compensación de movimiento de subpíxeles normalmente puede ser mucho más eficaz que la compensación de movimiento de píxeles enteros. Sin embargo, para algunos contenidos, como texturas con muy alta frecuencia o contenido de pantalla, la compensación de movimiento de subpíxeles puede no tener un mejor rendimiento o, en algunos casos, un rendimiento incluso peor. En dichos casos, puede ser mejor tener solo MV con precisión de píxel entero.

Como se describe en L. Guo, P. Yin, Y. Zheng, X. Lu, Q. Xu, J. Solé, "Adaptive motion vector resolution with implicit signaling," ICIP 2010: 2057-2060, se propuso una resolución de MV adaptable basada en residuos reconstruidos. Cuando la varianza del bloque de residuos reconstruido está por encima de un umbral, se utiliza la precisión del vector de movimiento de un cuarto de píxel. De lo contrario, se aplica la precisión del vector de movimiento de medio píxel. Como se describe en J. An, X. Li, X. Guo, S. Lei, "Progressive MV Resolution," JCTVC-F125, Torino, Italia, julio de 2011, la resolución de MV se determina de forma adaptable en base a la magnitud de la diferencia de MV señalizada. Como se describe en Y. Zhou, B. Li, J. Xu, G. J. Sullivan, B. Lin, "Motion Vector Resolution Control for Screen Content Coding", JCTVC-P0277, San José, EE. UU., enero de 2014, la información de la precisión del vector de movimiento se señala a nivel de fragmento.

Hasta ahora, los procedimientos de derivación de la precisión del vector de movimiento en el lado del descodificador no han demostrado ser muy eficaces, especialmente para la codificación del contenido de pantalla. Además, ningún procedimiento de derivación de la precisión del vector de movimiento en el lado del descodificador ha demostrado ser eficaz para todo tipo de contenido, incluido el contenido adquirido por la cámara, el contenido de la pantalla y otros tipos de contenido. Además, los procedimientos de precisión del vector de movimiento adaptables hasta ahora han demostrado no ser eficaces para la codificación del contenido de pantalla.

En algunos ejemplos, implementando las técnicas descritas en esta divulgación, se propone un procedimiento de derivación de la precisión del vector de movimiento en el lado del descodificador para el contenido de pantalla. En este ejemplo, la precisión del vector de movimiento puede depender del resultado de la coincidencia de la plantilla en el lado del descodificador. Cuando un resultado de la coincidencia de la plantilla de una posición de un píxel entero y el de su posición de subpíxel contiguo son bastante diferentes, la región relacionada puede considerarse como contenido de pantalla y el MV debe usarse con precisión de un píxel entero. De lo contrario, se utiliza la precisión del vector de movimiento de subpíxel. Para definir "bastante diferente", se pueden usar uno o más umbrales fijos o adaptables.

El descodificador de vídeo 30 puede, por ejemplo, descodificar datos de vídeo determinando una precisión del vector de movimiento basada en la coincidencia de plantillas. En un ejemplo de este tipo, el descodificador de vídeo 30 puede, para un bloque actual que se está codificando, identificar una posición de un píxel entero de un bloque contiguo ya codificado y, basándose en una ubicación de la posición de un píxel entero, aplicar una plantilla para determinar una pluralidad de posiciones de píxel entero. El descodificador de vídeo 30 también puede aplicar la plantilla a una pluralidad de posiciones de subpíxel para determinar una pluralidad de posiciones de subpíxel. La plantilla puede, por ejemplo, definir una forma, y el descodificador de vídeo 30 puede aplicar la plantilla a los datos de vídeo para determinar la pluralidad de posiciones de píxel entero localizando la pluralidad de posiciones de píxel entero basándose en una ubicación de la forma en relación con el bloque actual. De manera similar, el descodificador de vídeo 30 puede aplicar la plantilla a los datos de vídeo para determinar la pluralidad de posiciones de subpíxel localizando la pluralidad de posiciones de píxeles de subpíxel en base a una ubicación de la forma en relación con el bloque actual.

El descodificador de vídeo 30 puede comparar uno o más valores de los píxeles para la pluralidad de posiciones de píxel entero con uno o más valores de los píxeles para la pluralidad de posiciones de subpíxel y, basándose en la comparación, determinar una precisión del vector de movimiento para un vector de movimiento. El descodificador de vídeo 30 puede descodificar el bloque actual usando el vector de movimiento. El descodificador de vídeo 30 puede, por ejemplo, determinar el vector de movimiento usando un modo de fusión, un modo de AMVP o algún otro modo similar.

El descodificador de vídeo 30 puede determinar la precisión del vector de movimiento para el vector de movimiento comparando uno o más valores de los píxeles para la pluralidad de posiciones de píxel entero con uno o más valores de los píxeles para la pluralidad de posiciones de subpíxel para determinar un valor de diferencia que corresponde a una cantidad de diferencia en los valores de los píxeles entre los uno o más valores de los píxeles para la pluralidad de posiciones de píxel entero y los uno o más valores de los píxeles para la pluralidad de posiciones de subpíxel. En respuesta a que el valor de la diferencia sea mayor que un valor umbral, el descodificador de vídeo 30 determina que la precisión del vector de movimiento es la precisión de un píxel entero. En respuesta a que el valor de la diferencia sea menor que un valor umbral, el descodificador de vídeo 30 puede determinar que la precisión del vector de movimiento sea una precisión de subpíxel. El valor umbral puede ser un valor fijo, un valor adaptable o algún otro tipo de valor. Para comparar los uno o más valores de los píxeles de la pluralidad de posiciones de píxel entero con los uno o más valores de los píxeles de la pluralidad de posiciones de subpíxel, el descodificador de vídeo 30 puede, por ejemplo, determinar una suma de diferencias absolutas entre los uno o más valores de los píxeles de la pluralidad de posiciones de píxel entero y los uno o más valores de los píxeles de la pluralidad de posiciones de subpíxel.

De acuerdo con otras técnicas de esta divulgación, la precisión del vector de movimiento puede depender de las propiedades (como la nitidez, el gradiente o si se omite la transformada) de los bloques espacialmente contiguos, los bloques temporalmente contiguos o ambos. La información de precisión del vector de movimiento puede derivarse en el lado del descodificador. De forma adicional o alternativa, la precisión del vector de movimiento puede depender de la precisión del vector de movimiento de los bloques espacialmente contiguos, de los bloques temporalmente contiguos o de ambos.

El decodificador de vídeo 30 puede, por ejemplo, determinar la precisión de un vector de movimiento basándose en las propiedades del bloque contiguo. Los bloques contiguos pueden incluir, por ejemplo, al menos un bloque espacialmente contiguo y/o al menos un bloque temporalmente contiguo. Para un bloque actual que se codifica, el decodificador de vídeo 30 puede localizar uno o más bloques contiguos y determinar una propiedad de los uno o más bloques contiguos. La propiedad puede ser, por ejemplo, una o más de una nitidez de los uno o más bloques contiguos, un gradiente de los uno o más bloques contiguos, si uno o más bloques contiguos se ha codificado en un modo de omisión, y/o una precisión del vector de movimiento de los uno o más bloques contiguos. Basándose en la propiedad de los uno o más bloques contiguos, el decodificador de vídeo 30 puede determinar una precisión del vector de movimiento para un vector de movimiento y decodificar el bloque actual usando el vector de movimiento. El decodificador de vídeo 30 puede, por ejemplo, determinar sin señalización (por ejemplo, basándose en un contexto) qué propiedad o propiedades determinar, puede siempre determinar una propiedad o propiedades fijas, o puede recibir una indicación de qué propiedad o propiedades determinar.

En otra técnica de ejemplo de esta divulgación, se puede señalar en el flujo de bits un indicador sobre qué procedimiento o procedimientos de precisión del vector de movimiento se utilizan en el lado del decodificador. Por ejemplo, el indicador puede señalizarse en el flujo de bits directamente o derivarse de otra información codificada en el flujo de bits, como el tipo de fragmento y el nivel temporal.

El decodificador de vídeo 30 puede, por ejemplo, recibir en un flujo de bits de vídeo codificado, una indicación de un tipo de señalización de precisión del vector de movimiento y, basándose en el tipo de señalización de precisión del vector de movimiento, determinar una precisión del vector de movimiento para un bloque de datos de vídeo. El decodificador de vídeo 30 puede usar un vector de movimiento de la precisión del vector de movimiento determinada para localizar un bloque de referencia para el bloque de datos de vídeo. El tipo de señalización de precisión del vector de movimiento puede ser, por ejemplo, uno de (1) un tipo de coincidencia de la plantilla como se describe anteriormente, (2) un tipo basado en propiedades de bloque contiguo como se describe anteriormente, o (3) un tipo de señalización directa como se describirá más detalladamente a continuación.

El decodificador de vídeo 30 puede, por ejemplo, recibir la indicación en una cabecera de fragmento, un SPS, un PPS o en algún otro nivel. La indicación puede, por ejemplo, incluir un tipo de fragmento. En otras palabras, el decodificador de vídeo 30 puede determinar un tipo de fragmento para un fragmento particular y, en base a ese tipo de fragmento, puede determinar una precisión del vector de movimiento que se usará para decodificar bloques de ese fragmento. La indicación puede ser, por ejemplo, un nivel temporal de un fragmento. En otras palabras, el decodificador de vídeo 30 puede determinar un nivel temporal para un fragmento y, basándose en el nivel temporal del fragmento, determinar una precisión del vector de movimiento que se usarán para los bloques de decodificación del fragmento.

En otro ejemplo, la información de precisión del vector de movimiento puede señalizarse en el flujo de bits, tal como en el nivel de unidad de codificación más grande de LCU, el nivel de CU o el nivel de PU. En otras palabras, el codificador de vídeo 20 puede generar uno o más elementos de sintaxis para su inclusión en el flujo de bits de datos de vídeo codificados, y el decodificador de vídeo 30 puede analizar esos elementos de sintaxis para determinar la precisión del vector de movimiento para un bloque particular de datos de vídeo. Cuando se indica que una CU tiene los MV con precisión de número entero, todas las PU dentro de esta CU tienen una precisión del vector de movimiento de número entero.

En un ejemplo, para el modo de fusión/omisión, el decodificador de vídeo 30 puede redondear un vector de movimiento a una precisión de número entero solo cuando se realiza la compensación de movimiento. El MV sin redondear puede guardarse para la predicción del MV de bloques posteriores. Por ejemplo, el decodificador de vídeo 30 puede determinar un modo de codificación para un primer bloque, un modo de fusión o un modo de omisión, y determinar que una precisión del vector de movimiento para el primer bloque es una precisión de un píxel entero. El decodificador de vídeo 30 puede construir una lista de candidatos de fusión para el primer bloque que incluye al menos un candidato del vector de movimiento con precisión fraccionaria. El decodificador de vídeo 30 puede seleccionar el candidato del vector de movimiento con precisión fraccionaria para decodificar el primer bloque y redondear el candidato del vector de movimiento con precisión fraccionaria para determinar un vector de movimiento con precisión de un píxel entero. El decodificador de vídeo 30 puede localizar un bloque de referencia para el primer bloque usando el vector de movimiento con precisión de un píxel entero.

Para un segundo bloque (por ejemplo, un bloque codificado en base a la información del primer bloque), el decodificador de vídeo 30 puede añadir el candidato del vector de movimiento con precisión de número entero a una lista de candidatos (por ejemplo, una lista de candidatos de fusión o una lista de candidatos de AMVP) para el segundo bloque. En otros ejemplos, sin embargo, el decodificador de vídeo 30 puede añadir el candidato del vector de movimiento con precisión fraccionaria a una lista de candidatos para un segundo bloque.

Para el intermodo sin fusión/omisión, los predictores de MV se pueden redondear a una precisión de número entero, y la MVD se puede señalar con precisión de número entero para que el MV redondeado se pueda guardar para una predicción de MV de un bloque posterior. De forma alternativa o adicional, MV antes del redondeo puede guardarse

para la predicción de MV de un bloque posterior. En un ejemplo, para este caso, el redondeo se puede realizar solo para la compensación de movimiento. De forma alternativa o adicional, el MV redondeado se puede utilizar en la compensación de movimiento y se puede guardar para la predicción de MV de un bloque posterior.

Por ejemplo, el descodificador de vídeo 30 puede determinar que un modo de codificación para un primer bloque es distinto de un modo de fusión y determinar que una precisión del vector de movimiento para el primer bloque es la precisión de un píxel entero. El descodificador de vídeo 30 puede determinar un MVP con precisión fraccionaria para el primer bloque y redondear el MVP de precisión fraccionaria para determinar un MVP con precisión de un píxel entero para el primer bloque. El descodificador de vídeo 30 puede determinar una MVD para el primer bloque que es con precisión de un píxel entero. El descodificador de vídeo 30 puede determinar un vector de movimiento con precisión de un píxel entero basado en el MVP con precisión de un píxel entero y la MVD con precisión fraccionaria. El descodificador de vídeo 30 puede localizar un bloque de referencia para el primer bloque usando el vector de movimiento con precisión de un píxel entero.

El descodificador de vídeo 30 puede, por ejemplo, determinar el MVP de precisión fraccionaria para el primer bloque construyendo una lista de candidatos AMVP para el primer bloque. La lista de candidatos de AMVP puede incluir un candidato del vector de movimiento con precisión fraccionaria. El descodificador de vídeo 30 puede seleccionar el candidato del vector de movimiento con precisión fraccionaria como MVP de precisión fraccionaria para el primer bloque. El descodificador de vídeo 30 puede añadir el candidato del vector de movimiento con precisión fraccionaria a una lista de candidatos para un segundo bloque que se va a predecir utilizando información del primer bloque.

De forma adicional o alternativa, en un ejemplo, la información de precisión de la MVD se puede señalar, y siempre se puede usar el MV con precisión de subpíxel, en algunos ejemplos. La precisión de la MVD se puede señalar a nivel de LCU, a nivel de CU o a nivel de PU. En un ejemplo, cuando se indica que una PU (o CU) tiene una MVD con precisión de número entero, la PU (o todas las PU dentro de esta CU) puede tener una precisión de la MVD de número entero. Para las PU codificadas con AMVP, la MVD de las PU puede tener una precisión de píxel entero, mientras que el MV y el MV pronosticado de la PU pueden tener una precisión de subpíxel. Por lo tanto, añadir una MVD con precisión de número entero a un MVP con precisión de subpíxel da como resultado un vector de movimiento de subpíxel.

Por ejemplo, el descodificador de vídeo 30 puede determinar una precisión de la MVD para un primer bloque es la precisión de un píxel entero. El descodificador de vídeo 30 puede construir una lista de candidatos (por ejemplo, una lista de candidatos de AMVP) de los MVP para el primer bloque que incluye al menos un candidato del vector de movimiento con precisión fraccionaria. El descodificador de vídeo 30 puede seleccionar de la lista de candidatos el candidato del vector de movimiento con precisión fraccionaria y determinar un vector de movimiento con precisión de píxel fraccionario basado en el candidato del vector de movimiento con precisión fraccionaria y la MVD con precisión de un píxel entero. El descodificador de vídeo 30 puede localizar un bloque de referencia para el primer bloque utilizando el vector de movimiento con precisión de píxel fraccionario.

En otro ejemplo, el indicador de precisión del vector de movimiento se puede aplicar parcialmente a una LCU o una CU. Por ejemplo, el indicador de precisión de número entero de una CU no se aplica a sus PU que están codificadas con modos de codificación predefinidos, tal como fusión y omisión, o con particiones predefinidas, como particiones que no son de  $2N \times 2N$ , o con una herramienta de codificación especial, tal como omisión de transformada o sin residuos.

Por ejemplo, el descodificador de vídeo 30 puede determinar para los datos de vídeo una precisión del vector de movimiento por defecto y, en respuesta a una PU de los datos de vídeo que se codifican en un modo especial, localizar un bloque de referencia para la PU usando un vector de movimiento de la precisión del vector de movimiento por defecto. El modo especial puede ser, por ejemplo, uno o más de un modo de omisión, un modo de fusión de  $2N \times 2N$ , un modo de fusión, un modo de omisión de transformada o un modo de partición asimétrico. En respuesta a una segunda PU de los datos de vídeo que se codifica usando modos distintos a un modo especial, el descodificador de vídeo 30 puede determinar para la segunda PU de los datos de vídeo, una precisión del vector de movimiento señalizada y localizar un bloque de referencia para la segunda PU usando un vector de movimiento de la precisión del vector de movimiento señalizada. El descodificador de vídeo 30 puede determinar para una CU de los datos de vídeo, una precisión del vector de movimiento señalizada que es diferente de la precisión del vector de movimiento por defecto. La CU puede incluir, por ejemplo, la PU y/o la segunda PU. En un ejemplo, la precisión del vector de movimiento señalizada puede ser una precisión de un píxel entero mientras que la precisión del vector de movimiento por defecto es una precisión del vector de movimiento fraccionario. En otros ejemplos, la precisión del vector de movimiento por defecto puede ser una precisión del vector de movimiento fraccionario.

En un ejemplo, la información de precisión de MV/MVD se puede señalar solo para la PU o CU que tiene una MVD distinta de cero. Cuando no se señala la información de precisión de MV/MVD, se puede utilizar un MV de subpíxel para la PU o la CU. La información de precisión de MV/MVD se puede señalar después de la MVD de una PU o CU. Una MVD igual a cero puede usarse para significar que tanto el componente vertical de la MVD como los componentes horizontales de la MVD son iguales a 0.



Por ejemplo, para un bloque actual de datos de vídeo, el descodificador de vídeo 30 puede recibir un valor de MVD y, en respuesta a que el valor de MVD sea igual a cero, determinar un vector de movimiento para el bloque actual que tiene una precisión del vector de movimiento de subpíxel. El valor de la MVD igual a cero puede indicar que tanto un componente x del valor de MVD como un componente y del valor de MVD son iguales a cero. Para un segundo bloque actual de datos de vídeo, el descodificador de vídeo 30 puede recibir un segundo valor de MVD y, en respuesta a que el segundo valor de MVD sea un valor distinto de cero, recibir una indicación de una precisión del vector de movimiento para un segundo vector de movimiento del segundo bloque actual. El descodificador de vídeo 30 puede ubicar, en una imagen de referencia, un bloque de referencia para el segundo bloque actual usando el segundo vector de movimiento. Para el segundo bloque actual, el descodificador de vídeo 30 puede recibir la indicación de la precisión del vector de movimiento después de recibir el segundo valor de MVD.

Cuando la información de precisión de MV/MVD se señala en el nivel de PU, la información de precisión de MV/MVD puede no señalizarse si una o más (por ejemplo, cualquiera) de las siguientes condiciones son verdaderas: (1) la PU está codificada con el modo de fusión/omisión, (2) la PU está codificada con el modo de AMVP, y la MVD en cada dirección de predicción de la PU es igual a cero, o (3) de forma adicional o alternativa, si una CU pudiera contener PU intracodificadas y PU intercodificadas juntas, lo que no está permitido en HEVC, y cuando la PU está intracodificada, se omite la señalización de la información de precisión de MV/MVD a nivel de PU.

El descodificador de vídeo 30 puede, por ejemplo, recibir, para un primer bloque de datos de vídeo (por ejemplo, una primera PU), la primera información de precisión del vector de movimiento. En respuesta a un segundo bloque de datos de vídeo que cumple con una condición, el descodificador de vídeo 30 puede determinar la segunda información de los vectores de movimiento para que se corresponda con una precisión por defecto. En un ejemplo, la condición puede ser que el segundo bloque se codifique utilizando el modo de fusión o el modo de omisión. En otro ejemplo, la condición puede ser que el segundo bloque se codifique utilizando el modo de AMVP y una MVD para cada dirección de predicción del segundo bloque que sea igual a cero. La precisión predeterminada puede, por ejemplo, ser una precisión fraccionaria en algunos ejemplos o una precisión de número entero en otros ejemplos. La primera y la segunda información de precisión del vector de movimiento pueden ser, por ejemplo, una o ambas de una precisión del vector de movimiento o precisión de la MVD.

Cuando la información de precisión de MV/MVD se señala en el nivel de CU, la información de precisión de MV/MVD puede no señalizarse si una (y posiblemente una o más) de las siguientes condiciones es verdadera para todas las PU dentro de la CU: (1) la PU está codificada internamente, (2) la PU está codificada con el modo de fusión/omisión, o (3) la PU está codificada con el modo de AMVP, y la MVD en cada dirección de predicción de la PU es igual a cero. De forma adicional o alternativa, cuando la información de precisión del vector de movimiento no está señalizada, se puede usar una precisión del vector de movimiento por defecto, tal como la precisión del vector de movimiento de número entero, para la PU o CU.

El descodificador de vídeo 30 puede, por ejemplo, recibir, para una primera CU de datos de vídeo, la primera información de precisión del vector de movimiento y, en respuesta a una segunda CU de los datos de vídeo que cumplen con una condición, determinar la segunda información de los vectores de movimiento para que se corresponda con una precisión por defecto. La condición puede ser, por ejemplo, que todas las PU dentro de la CU estén codificadas internamente, todas las PU dentro de la CU estén codificadas utilizando el modo de fusión o el modo de omisión, todas las PU dentro de la CU estén codificadas utilizando AMVP y una MVD para cada dirección de todas las PU que son igual a cero. La precisión por defecto puede ser, por ejemplo, precisión fraccionaria o puede no tener precisión. Por ejemplo, si un bloque es intrapredicho, entonces el bloque no tiene un vector de movimiento asociado y, así pues, no tiene una precisión del vector de movimiento asociada. La primera y la segunda información de precisión del vector de movimiento pueden incluir, por ejemplo, una o ambas de precisión del vector de movimiento o precisión de la MVD.

Cuando la PU codificada en AMVP actual se señala/deriva como con precisión del vector de movimiento de píxel entero, uno o más (y en algunos ejemplos, todos) candidatos de MV de bloques contiguos espaciales, bloques contiguos temporales o ambos pueden redondearse a una precisión de píxel entero antes de depurarse en el proceso de generación de la lista de AMVP. Cuando se señala/deriva un MV de píxel entero para utilizarse en una fusión actual, una CU/PU codificada con omisión, o ambos, uno o más (y en algunos ejemplos, todos) candidatos de MV de bloques contiguos temporales espaciales, bloques contiguos temporales, o ambos, se pueden redondear a una precisión de píxel entero antes de depurarse en el proceso de generación de la lista de fusión.

Por ejemplo, el descodificador de vídeo 30 puede identificar uno o más candidatos del vector de movimiento para su inclusión en una lista de candidatos (por ejemplo, una lista de candidatos de fusión o una lista de candidatos de AMVP) para un bloque. El uno o más candidatos del vector de movimiento pueden incluir, por ejemplo, uno o más candidatos contiguos espaciales y/o uno o más candidatos contiguos temporales. El uno o más candidatos del vector de movimiento pueden incluir al menos un candidato del vector de movimiento con precisión fraccionaria. En respuesta a una precisión del vector de movimiento para el bloque que es una precisión de un píxel entero, el descodificador de vídeo 30 puede redondear el candidato o los candidatos del vector de movimiento para determinar uno o más candidatos del vector de movimiento con precisión de número entero. El descodificador de vídeo 30 puede realizar una operación de depuración en uno o más candidatos del vector de movimiento con precisión de número entero.

En un ejemplo, el indicador de precisión del vector de movimiento puede usarse, o usarse condicionalmente, como contextos CABAC de otros elementos de sintaxis. Es decir, se pueden usar diferentes modelos de contexto, dependiendo del indicador de precisión del vector de movimiento, para codificar determinado elemento de sintaxis. En un ejemplo, cuando se codifica un índice de candidatos de AMVP para un bloque tal como la PU, el(los) indicador(es) de precisión del vector de movimiento de una PU o una CU asociada o bloques espacialmente contiguos o bloques temporalmente contiguos se utiliza(n) como contexto(s) de codificación de CABAC. De forma adicional o alternativa, en algunos ejemplos, la probabilidad inicializada de que el índice de candidatos de AMVP sea igual a 0 puede establecerse cerca de 1 cuando el indicador de precisión del vector de movimiento indica la precisión del vector de movimiento de píxel entero. De forma adicional o alternativa, en algunos casos, tal como solo en fragmentos B, o solo cuando el fragmento está en un determinado nivel temporal, o cuando el parámetro de cuantificación es mayor que un umbral predefinido, el indicador de precisión del vector de movimiento puede usarse como contextos de CABAC para otros elementos de sintaxis, tal como el índice de candidatos de AMVP.

Uno o más de estos ejemplos se pueden combinar. Por ejemplo, en la práctica, cualquier combinación de cualquier parte del ejemplo puede usarse como un nuevo ejemplo. Además, a continuación se analizan subejemplos de los ejemplos anteriores.

Algunos ejemplos se refieren a la derivación de la precisión del vector de movimiento en el lado del descodificador para el contenido de la pantalla. En un ejemplo, se puede utilizar la coincidencia de plantillas en forma de L u otras formas en muestras reconstruidas. La precisión del vector de movimiento puede basarse en la diferencia entre el resultado de la coincidencia de la plantilla, tal como SAD, de una posición de un píxel entero y el resultado de la coincidencia de su posición de subpíxel contigua. Por ejemplo, cuando el resultado de la coincidencia de la posición de píxel entero es mucho más bajo, se aplica la precisión de píxel entero. De otro modo, se aplica la precisión de subpíxel. Para definir "mucho más bajo", se puede utilizar un umbral. En la práctica, se pueden utilizar un umbral fijo, un umbral adaptable o ambos. En el caso de un umbral adaptable, el umbral adaptable puede señalizarse en el flujo de bits o derivarse en base a otra información, como el tipo de bloque, o QP, señalizado en el flujo de bits. Además, también se puede definir un umbral para un caso "mucho más alto". En consecuencia, cuando el resultado de la coincidencia de la posición de número entero menos el de la posición del subpíxel contiguo es mayor que el umbral "mucho más alto", se puede usar la precisión de un cuarto de píxel. Cuando la diferencia de coincidencia se encuentra entre los umbrales de "mucho más bajo" y "mucho más alto", se puede usar precisión de medio píxel. De forma alternativa o adicional, en el ejemplo anterior se puede usar otro procedimiento de coincidencia de la plantilla, tal como la coincidencia de plantilla bidireccional basada en espejo.

En otro ejemplo, la información de precisión del vector de movimiento puede derivarse en el lado del descodificador basándose en la propiedad de los bloques contiguos espacial o temporalmente, como el gradiente, la nitidez o si la transformación se omite para los bloques. La información de umbral se puede señalar en un flujo de bits, derivar del flujo de bits o ambos.

Algunos ejemplos se refieren a la señalización de indicadores. Para ajustarse de forma adaptable a diferentes contenidos, se puede utilizar una combinación de diferentes procedimientos de derivación de la precisión del vector de movimiento en el lado del descodificador (DMPD). Para señalar qué procedimiento o procedimientos están en uso, se puede señalar un indicador en un flujo de bits. En un ejemplo, el indicador se puede señalar a nivel de fragmento o superior para indicar explícitamente al descodificador qué procedimiento o procedimientos de DMPD se utilizarán. En otro ejemplo, la utilización de algunos procedimientos de DMPD se señala en un flujo de bits, mientras que la utilización de otros procedimientos de DMPD se deriva en base a otra información como, por ejemplo, el tipo de fragmento y el nivel temporal del fragmento, en el flujo de bits.

Algunos ejemplos se refieren a la precisión del vector de movimiento adaptable señalizada. En un ejemplo de este tipo, la precisión del vector de movimiento se puede señalar en un flujo de bits, tal como en el nivel de LCU, CU o PU. Se puede usar un indicador/valor para indicar la precisión del vector de movimiento, tal como la precisión de número entero, precisión de medio píxel, precisión de un cuarto de píxel u otras precisiones. Cuando la precisión del vector de movimiento se señala para un bloque o una región/fragmento, todos los bloques más pequeños dentro de este bloque/región/fragmento pueden compartir la misma precisión del vector de movimiento. Además, la información de MVD también puede señalizarse en la precisión señalizada. Antes de la compensación de movimiento, el MV (predicador de MV + MVD) puede redondearse a la precisión señalizada. El redondeo puede ser hacia infinito positivo, infinito negativo, cero o infinito (el valor negativo se redondea a infinito negativo mientras que el valor positivo se redondea a infinito positivo). De forma alternativa o adicional, el predictor de MV puede redondearse como se ha mencionado anteriormente y luego formar el MV para un bloque. Después de la compensación de movimiento, el MV del bloque se guarda para la predicción del MV de bloques posteriores. Al guardar el MV, el MV redondeado se puede guardar, por ejemplo, para usarlo más tarde como candidato de fusión o candidato de AMVP para un bloque descodificado posteriormente. De forma alternativa o adicional, el MV no redondeado se puede guardar en lugar del vector de movimiento redondeado, lo que potencialmente puede mantener el campo de movimiento más exacto.

En otro ejemplo, la información de precisión del vector de movimiento no se señala para el modo de omisión, el modo de fusión de 2Nx2N o ambos. En un ejemplo de este tipo, la información de precisión del vector de movimiento también

podría no señalizarse para una PU fusionada. De forma adicional o alternativa, las PU que están codificadas en modos de codificación especiales, tal como el modo de fusión y el modo de omisión, o con particiones especiales, tal como las particiones asimétricas, o con una profundidad de transformada especial o con omisión de transformada, pueden mantener la precisión del vector de movimiento por defecto, como por ejemplo, cuarto de pel, incluso cuando el MV con precisión de número entero se señala en su nivel de CU. De forma adicional o alternativa, otra información codificada, tal como el nivel temporal, el QP, la profundidad de CU, también puede considerarse como un modo de codificación especial o una herramienta de codificación especial.

Cuando se codifica por entropía la información de precisión del vector de movimiento con CABAC, se pueden utilizar contextos distintos de la información de precisión del vector de movimiento en bloques/CU espacialmente contiguos para guardar el búfer de línea, tal como la profundidad de CU, la partición de PU, el tamaño de bloque, el nivel temporal, etcétera.

La FIG. 8 es un diagrama de flujo que ilustra un procedimiento de ejemplo para codificar datos de vídeo de acuerdo con las técnicas descritas en esta divulgación. Las técnicas de la FIG. 8 se describirán en referencia a un codificador de vídeo, tal como el codificador de vídeo 20. El codificador de vídeo 20 determina que una precisión del vector de movimiento para un primer bloque es la precisión de un píxel entero (202). El codificador de vídeo 20 construye una lista de candidatos de fusión para el primer bloque (204). Como parte de la construcción de la lista de candidatos de fusión, el codificador de vídeo 20 puede añadir un candidato del vector de movimiento con precisión fraccionaria a la lista de candidatos de fusión. Por tanto, la lista de candidatos de fusión puede incluir un candidato del vector de movimiento con precisión fraccionaria. El codificador de vídeo 20 selecciona el candidato del vector de movimiento con precisión fraccionaria para codificar el primer bloque (206). Para codificar el primer bloque usando un modo de fusión, el codificador de vídeo 20 redondea el candidato del vector de movimiento con precisión fraccionaria para determinar un vector de movimiento con precisión de un píxel entero para el primer bloque (208). El codificador de vídeo 20 localiza un bloque de referencia para el primer bloque usando el vector de movimiento con precisión de un píxel entero (210). El codificador de vídeo 20 codifica el primer bloque en base al bloque de referencia (212).

La FIG. 9 es un diagrama de flujo que ilustra un procedimiento de ejemplo para descodificar datos de vídeo de acuerdo con las técnicas descritas en esta divulgación. Las técnicas de la FIG. 9 se describirán en referencia a un descodificador de vídeo, tal como el descodificador de vídeo 30. El descodificador de vídeo 30 determina que un modo de codificación para un primer bloque es el modo de fusión (220). El descodificador de vídeo 30 determina que una precisión del vector de movimiento para el primer bloque es la precisión de un píxel entero (222). El descodificador de vídeo 30 construye una lista de candidatos de fusión para el primer bloque (224). El descodificador de vídeo 30 construye la lista de candidatos de fusión añadiendo un candidato del vector de movimiento con precisión fraccionaria a la lista de candidatos de fusión. El descodificador de vídeo 30 selecciona el candidato del vector de movimiento con precisión fraccionaria para descodificar el primer bloque (226). En respuesta a la selección del vector de movimiento con precisión fraccionaria, el descodificador de vídeo 30 redondea el candidato del vector de movimiento con precisión fraccionaria para determinar un vector de movimiento con precisión de un píxel entero para el primer bloque (228). El descodificador de vídeo 30 localiza un bloque de referencia para el primer bloque usando el vector de movimiento con precisión de un píxel entero (230). El descodificador de vídeo 30 descodifica el primer bloque en base al bloque de referencia (232).

La FIG. 10 es un diagrama de flujo que ilustra un procedimiento de ejemplo para codificar datos de vídeo de acuerdo con las técnicas descritas en esta divulgación. Las técnicas de la FIG. 10 se describirán en referencia al descodificador de vídeo 30; sin embargo, muchas de las técnicas también se pueden realizar mediante un codificador de vídeo, tal como el codificador de vídeo 20. Para un bloque actual que se codifica en una imagen actual, el descodificador de vídeo 30 identifica una posición de un píxel entero de un bloque contiguo ya codificado (234). Basándose en una ubicación de la posición de píxel entero, el descodificador de vídeo 30 aplica una plantilla para determinar una pluralidad de posiciones de píxel entero en la imagen (236). El descodificador de vídeo 30 aplica la plantilla a una pluralidad de posiciones de subpíxel para determinar una pluralidad de posiciones de subpíxel en la imagen (238). El descodificador de vídeo 30 compara uno o más valores de los píxeles para la pluralidad de posiciones de píxel entero con uno o más valores de los píxeles para la pluralidad de posiciones de subpíxel (240). Basándose en la comparación, el descodificador de vídeo 30 determina una precisión del vector de movimiento para un vector de movimiento (242). El descodificador de vídeo 30 codifica el bloque actual usando el vector de movimiento con la precisión del vector de movimiento determinada (242).

La FIG. 11 es un diagrama de flujo que ilustra un procedimiento de ejemplo para codificar datos de vídeo de acuerdo con las técnicas descritas en esta divulgación. Las técnicas de la FIG. 11 se describirán en referencia al descodificador de vídeo 30; sin embargo, muchas de las técnicas también se pueden realizar mediante un codificador de vídeo, tal como el codificador de vídeo 20. Para un bloque actual que se está codificando, el descodificador de vídeo 30 localiza uno o más bloques contiguos (246). El descodificador de vídeo 30 determina una propiedad de los uno o más bloques contiguos (248). Basándose en la propiedad de los uno o más bloques contiguos, el descodificador de vídeo 30 determina una precisión del vector de movimiento para un vector de movimiento (250). El descodificador de vídeo 30 codifica el bloque actual usando el vector de movimiento con la precisión del vector de movimiento determinada.

La FIG. 12 es un diagrama de flujo que ilustra un procedimiento de ejemplo para codificar datos de vídeo de acuerdo con las técnicas descritas en esta divulgación. Las técnicas de la FIG. 12 se describirán en referencia al descodificador de vídeo 30; sin embargo, muchas de las técnicas también se pueden realizar mediante un codificador de vídeo, tal como el codificador de vídeo 20. El descodificador de vídeo 30 determina que un modo de codificación para un primer bloque es el modo de fusión (254). El descodificador de vídeo 30 determina que una precisión del vector de movimiento para el primer bloque es la precisión de un píxel entero (256). El descodificador de vídeo 30 construye una lista de candidatos de fusión para el primer bloque, en la que la lista de candidatos de fusión comprende un candidato del vector de movimiento con precisión fraccionaria (258). El descodificador de vídeo 30 selecciona el candidato del vector de movimiento con precisión fraccionaria para descodificar el primer bloque (260). El descodificador de vídeo 30 redondea el candidato del vector de movimiento con precisión fraccionaria para determinar un vector de movimiento con precisión de un píxel entero (262). El descodificador de vídeo 30 localiza un bloque de referencia para el primer bloque usando el vector de movimiento con precisión de un píxel entero.

La FIG. 13 es un diagrama de flujo que ilustra un procedimiento de ejemplo para codificar datos de vídeo de acuerdo con las técnicas descritas en esta divulgación. Las técnicas de la FIG. 13 se describirán en referencia al descodificador de vídeo 30; sin embargo, muchas de las técnicas también se pueden realizar mediante un codificador de vídeo, tal como el codificador de vídeo 20. Para un bloque actual de datos de vídeo, el descodificador de vídeo 30 determina un valor de diferencia del vector de movimiento (266). En respuesta a que el valor de la diferencia del vector de movimiento sea igual a cero, el descodificador de vídeo 30 determina que un vector de movimiento para el bloque actual tiene una precisión del vector de movimiento de subpíxel (268).

La FIG. 14 es un diagrama de flujo que ilustra un procedimiento de ejemplo para codificar datos de vídeo de acuerdo con las técnicas descritas en esta divulgación. Las técnicas de la FIG. 14 se describirán en referencia al descodificador de vídeo 30; sin embargo, muchas de las técnicas también se pueden realizar mediante un codificador de vídeo, tal como el codificador de vídeo 20. El descodificador de vídeo 30 determina una precisión del vector de movimiento por defecto (270) para los datos de vídeo. En respuesta a una PU de los datos de vídeo que se codifica en un modo especial, el descodificador de vídeo 30 localiza un bloque de referencia para la PU usando un vector de movimiento de la precisión del vector de movimiento por defecto (272).

La FIG. 15 es un diagrama de flujo que ilustra un procedimiento de ejemplo para codificar datos de vídeo de acuerdo con las técnicas descritas en esta divulgación. Las técnicas de la FIG. 15 se describirán en referencia al descodificador de vídeo 30; sin embargo, muchas de las técnicas también se pueden realizar mediante un codificador de vídeo, tal como el codificador de vídeo 20. El descodificador de vídeo 30 identifica uno o más candidatos del vector de movimiento para su inclusión en una lista de candidatos para un bloque, en el que los uno o más candidatos del vector de movimiento comprenden al menos un candidato del vector de movimiento con precisión fraccionaria (274). En respuesta a una precisión del vector de movimiento para el bloque que es una precisión de un píxel entero, el descodificador de vídeo 30 redondea el candidato o los candidatos del vector de movimiento para determinar uno o más candidatos del vector de movimiento con precisión de número entero (276). Después de redondear los uno o más candidatos del vector de movimiento, el descodificador de vídeo 30 realiza una operación de depuración en los uno o más candidatos del vector de movimiento con precisión de número entero (278).

La FIG. 16 es un diagrama de flujo que ilustra un procedimiento de ejemplo para codificar datos de vídeo de acuerdo con las técnicas descritas en esta divulgación. Las técnicas de la FIG. 16 se describirán en referencia al descodificador de vídeo 30; sin embargo, muchas de las técnicas también se pueden realizar mediante un codificador de vídeo, tal como el codificador de vídeo 20. El descodificador de vídeo 30 determina una precisión por defecto (280) para un primer bloque de datos de vídeo y un segundo bloque de datos de vídeo. El descodificador de vídeo 30 determina, para el primer bloque de datos de vídeo, la primera información de precisión del vector de movimiento (282). En respuesta al segundo bloque de datos de vídeo que cumple con una condición, se determina una segunda información del vector de movimiento para que se corresponda con la precisión por defecto (284). El primer y segundo bloques pueden ser, por ejemplo, la primera y segunda PU o la primera y segunda CU.

La FIG. 17 es un diagrama de flujo que ilustra un procedimiento de ejemplo para codificar datos de vídeo de acuerdo con las técnicas descritas en esta divulgación. Las técnicas de la FIG. 17 se describirán en referencia al descodificador de vídeo 30; sin embargo, muchas de las técnicas también se pueden realizar mediante un codificador de vídeo, tal como el codificador de vídeo 20. El descodificador de vídeo 30 determina una precisión de diferencia del vector de movimiento para un primer bloque es la precisión de un píxel entero (286). El descodificador de vídeo 30 construye una lista de candidatos de predictores del vector de movimiento para el primer bloque, en la que la lista de candidatos comprende un candidato del vector de movimiento con precisión fraccionaria (288). El descodificador de vídeo 30 selecciona de la lista de candidatos el candidato del vector de movimiento con precisión fraccionaria (290). El descodificador de vídeo 30 determina un vector de movimiento con precisión de píxel fraccionario basándose en el candidato del vector de movimiento con precisión fraccionaria y la diferencia del vector de movimiento con precisión de un píxel entero (292). El descodificador de vídeo 30 localiza un bloque de referencia para el primer bloque usando el vector de movimiento con precisión de píxel fraccionario (294).

Para facilitar la explicación, las técnicas de las FIG. 8 a 17 se han presentado por separado, pero se contempla que las técnicas descritas se puedan utilizar conjuntamente. Además, se contempla que porciones de algunas técnicas se puedan usar en combinación con porciones de otras técnicas.

En uno o más ejemplos, las funciones descritas se pueden implementar en hardware, software, firmware o en cualquier combinación de los mismos. Si se implementan en software, las funciones pueden almacenarse en, o transmitirse por, un medio legible por ordenador como una o más instrucciones o código, y ejecutarse mediante una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador que corresponden a un medio tangible, tales como unos medios de almacenamiento de datos, o unos medios de comunicación que incluyen cualquier medio que facilita la transferencia de un programa informático de un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador pueden corresponder, en general, a (1) unos medios de almacenamiento tangibles legibles por ordenador que no son transitorios o a (2) un medio de comunicación tal como una señal u onda portadora. Los medios de almacenamiento de datos pueden ser unos medios cualesquiera disponibles a los que se puede acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, un código y/o unas estructuras de datos para la implementación de las técnicas descritas en la presente divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

A modo de ejemplo, y no de limitación, dichos medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento en disco óptico, almacenamiento en disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que se puede usar para almacenar un código de programa deseado en forma de instrucciones o estructuras de datos y al que se puede acceder mediante un ordenador. Además, cualquier conexión recibe apropiadamente la denominación de medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, un servidor u otra fuente remota usando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o unas tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas están incluidos en la definición de medio. Sin embargo, se debe entender que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales ni otros medios transitorios, sino que, en su lugar, se dirigen a medios de almacenamiento tangibles no transitorios. Los discos, como se usan en el presente documento, incluyen el disco compacto (CD), el disco láser, el disco óptico, el disco versátil digital (DVD), el disco flexible y el disco Blu-ray, donde algunos discos reproducen habitualmente datos magnéticamente y otros discos reproducen datos ópticamente con láseres. Combinaciones de lo anterior también se deben incluir dentro del alcance de los medios legibles por ordenador.

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), matrices lógicas programables por campo (FPGA) u otros circuitos lógicos integrados o discretos equivalentes pueden ejecutar las instrucciones. En consecuencia, el término "procesador", como se usa en el presente documento, se puede referir a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento se puede proporcionar dentro de módulos de hardware y/o de software dedicados configurados para codificar y descodificar, o incorporar en un códec combinado. Asimismo, las técnicas se podrían implementar por completo en uno o más circuitos o elementos lógicos.

Las técnicas de esta divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluidos un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En esta divulgación se describen diversos componentes, módulos o unidades para destacar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no se requiere necesariamente su realización mediante diferentes unidades de hardware. En cambio, como se describe anteriormente, diversas unidades se pueden combinar en una unidad de hardware de códec o proporcionar mediante un grupo de unidades de hardware interoperativas, que incluyen uno o más procesadores descritos anteriormente, junto con software y/o firmware adecuados.

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

**REIVINDICACIONES**

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

5 determinar (254) que un modo de codificación para un primer bloque es el modo de fusión, en el que, en el modo de fusión, se construye una lista de candidatos de fusión que comprende candidatos del vector de movimiento basándose en bloques ya codificados;  
 10 determinar (256) que una precisión del vector de movimiento para el primer bloque es una precisión de un píxel entero;  
 construir (268) una lista de candidatos de fusión para el primer bloque, para incluir candidatos del vector de movimiento basándose en bloques ya codificados, en el que la lista de candidatos de fusión comprende un candidato del vector de movimiento con precisión fraccionaria;  
 15 seleccionar (260) el candidato del vector de movimiento con precisión fraccionaria;  
 en respuesta a la selección del candidato del vector de movimiento con precisión fraccionaria, redondear (262) el vector de movimiento con precisión fraccionaria candidato para determinar un vector de movimiento con precisión de un píxel entero para el primer bloque;  
 20 localizar (264) un bloque de referencia para el primer bloque usando el vector de movimiento con precisión de un píxel entero; y  
 decodificar el primer bloque basándose en el bloque de referencia.

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

determinar que un modo de codificación para un segundo bloque es distinto del modo de fusión;  
 determinar que una precisión del vector de movimiento para el segundo bloque es una precisión de un píxel entero;  
 25 determinar un predictor del vector de movimiento con precisión fraccionaria para el segundo bloque;  
 redondear el predictor del vector de movimiento con precisión fraccionaria para determinar un predictor del vector de movimiento con precisión de un píxel entero para el segundo bloque;  
 determinar una diferencia del vector de movimiento para el segundo bloque, en el que la diferencia del vector de movimiento comprende una diferencia con precisión de un píxel entero;  
 30 determinar un vector de movimiento con precisión de un píxel entero para el segundo bloque en base al predictor del vector de movimiento con precisión de un píxel entero y la diferencia del vector de movimiento; y  
 localizar un bloque de referencia para el segundo bloque utilizando el vector de movimiento con precisión de un píxel entero.

3. Un procedimiento de codificación de datos de vídeo, el procedimiento que comprende:

35 determinar codificar un primer bloque en el modo de fusión, en el que, en el modo de fusión, se construye una lista de candidatos de fusión que comprende candidatos del vector de movimiento basándose en bloques ya codificados;  
 determinar que una precisión del vector de movimiento para un primer bloque es una precisión de un píxel entero;  
 40 construir una lista de candidatos de fusión para el primer bloque, para incluir candidatos del vector de movimiento basándose en bloques ya codificados, en el que la lista de candidatos de fusión comprende un candidato del vector de movimiento con precisión fraccionaria;  
 seleccionar el candidato del vector de movimiento con precisión fraccionaria;  
 redondear el candidato del vector de movimiento con precisión fraccionaria para determinar un vector de movimiento con precisión de un píxel entero para el primer bloque;  
 45 localizar un bloque de referencia para el primer bloque usando el vector de movimiento con precisión de un píxel entero; y  
 codificar el primer bloque basándose en el bloque de referencia.

4. El procedimiento de la reivindicación 1, en el que determinar que la precisión del vector de movimiento para el primer bloque es la precisión de un píxel entero comprende recibir en los datos de vídeo codificados una indicación de que la precisión del vector de movimiento para el primer bloque es una precisión de un píxel entero, o el procedimiento de la reivindicación 3, que comprende además:  
 50 generar para su inclusión en los datos de vídeo codificados, una indicación de que una precisión del vector de movimiento para el primer bloque es una precisión de un píxel entero.

5. El procedimiento de la reivindicación 1 o la reivindicación 3, en el que el modo de fusión comprende un modo de omisión.

6. El procedimiento de la reivindicación 3, que comprende además:

60 determinar que una precisión del vector de movimiento para un segundo bloque es una precisión de un píxel entero;  
 para codificar el segundo bloque en un modo diferente al modo de fusión, determinar un predictor del vector de movimiento con precisión fraccionaria para el segundo bloque;  
 redondear el predictor del vector de movimiento con precisión fraccionaria para determinar un predictor del vector de movimiento con precisión de un píxel entero para el segundo bloque;

determinar una diferencia del vector de movimiento para el segundo bloque, en el que la diferencia del vector de movimiento comprende una diferencia con precisión de un píxel entero;  
 determinar un vector de movimiento con precisión de un píxel entero para el segundo bloque en base al predictor del vector de movimiento con precisión de un píxel entero y la diferencia del vector de movimiento; y  
 5 localizar un bloque de referencia para el segundo bloque utilizando el vector de movimiento con precisión de un píxel entero;  
 codificar el primer bloque basándose en el bloque de referencia.

7. El procedimiento de la reivindicación 2 o la reivindicación 6, en el que determinar el predictor del vector de movimiento con precisión fraccionaria para el segundo bloque comprende:  
 10 construir una lista de candidatos de predicción avanzada del vector de movimiento, AMVP, para el segundo bloque, en la que la lista de candidatos de AMVP comprende un segundo candidato del vector de movimiento con precisión fraccionaria;  
 seleccionar el segundo candidato del vector de movimiento con precisión fraccionaria como el predictor del vector de movimiento con precisión fraccionaria para el segundo bloque,  
 15 el procedimiento preferentemente que comprende además:  
 añadir el vector de movimiento con precisión de número entero a una lista de candidatos para un segundo bloque.

8. El procedimiento de la reivindicación 1 o la reivindicación 3, que comprende además:  
 20 añadir el candidato del vector de movimiento con precisión fraccionaria a una lista de candidatos para un segundo bloque, en el que la lista de candidatos para el segundo bloque comprende preferentemente una lista de candidatos de predicción avanzada del vector de movimiento, AMVP.

9. Un dispositivo para decodificación de vídeo, el dispositivo que comprende:

25 una memoria configurada para almacenar datos de vídeo;  
 un decodificador de vídeo que comprende uno o más procesadores configurados para:  
 determinar que un modo de codificación para un primer bloque es el modo de fusión, en el que, en el modo de fusión, se construye una lista de candidatos de fusión que comprende candidatos del vector de movimiento basándose en  
 30 bloques ya codificados;  
 determinar que una precisión del vector de movimiento para el primer bloque es una precisión de un píxel entero;  
 construir una lista de candidatos de fusión para el primer bloque basándose en datos de vídeo almacenados en la memoria, que incluya candidatos del vector de movimiento basados en bloques ya codificados, en el que la lista de candidatos de fusión comprende un candidato del vector de movimiento con precisión fraccionaria;  
 35 seleccionar el candidato del vector de movimiento con precisión fraccionaria;  
 redondear el candidato del vector de movimiento con precisión fraccionaria para determinar un vector de movimiento con precisión de un píxel entero para el primer bloque;  
 localizar un bloque de referencia para el primer bloque usando el vector de movimiento con precisión de un píxel entero; y  
 40 decodificar el primer bloque basándose en el bloque de referencia.

10. El dispositivo de la reivindicación 9, en el que los uno o más procesadores están configurados además para determinar que la precisión del vector de movimiento para el primer bloque es una precisión de un píxel entero, los uno o más procesadores están configurados además para recibir en los datos de vídeo codificados una indicación de  
 45 que la precisión del vector de movimiento para el primer bloque es una precisión de un píxel entero.

11. El dispositivo de la reivindicación 9, en el que los uno o más procesadores están configurados además para:

determinar que un modo de codificación para un segundo bloque es distinto del modo de fusión;  
 50 determinar que una precisión del vector de movimiento para el segundo bloque es una precisión de un píxel entero;  
 determinar un predictor del vector de movimiento con precisión fraccionaria para el segundo bloque;  
 redondear el predictor del vector de movimiento con precisión fraccionaria para determinar un predictor del vector de movimiento con precisión de un píxel entero para el segundo bloque;  
 determinar una diferencia del vector de movimiento para el segundo bloque, en el que la diferencia del vector de movimiento comprende una diferencia con precisión de un píxel entero;  
 55 determinar un vector de movimiento con precisión de un píxel entero para el segundo bloque en base al predictor del vector de movimiento con precisión de un píxel entero y la diferencia del vector de movimiento;  
 localizar un bloque de referencia para el segundo bloque usando el vector de movimiento con precisión de un píxel entero.

12. El dispositivo de la reivindicación 11, en el que para determinar el predictor del vector de movimiento con precisión fraccionaria para el segundo bloque, los uno o más procesadores están configurados además para:

65 construir una lista de candidatos de predicción avanzada del vector de movimiento, AMVP, para el segundo bloque, en la que la lista de candidatos de AMVP comprende un segundo candidato del vector de movimiento con precisión fraccionaria;

seleccionar el segundo candidato del vector de movimiento con precisión fraccionaria como el predictor del vector de movimiento con precisión fraccionaria para el segundo bloque, en el que los uno o más procesadores están configurados preferentemente además para:

añadir el vector de movimiento con precisión de número entero a una lista de candidatos para un segundo bloque.

5 13. El dispositivo de la reivindicación 9, en el que los uno o más procesadores están configurados además para: añadir el candidato del vector de movimiento con precisión fraccionaria a una lista de candidatos para un segundo bloque, en el que la lista de candidatos para el segundo bloque comprende preferentemente una lista de candidatos de predicción avanzada del vector de movimiento, AMVP.

10 14. Un medio de almacenamiento legible por ordenador que almacena instrucciones que, cuando son ejecutadas por uno o más procesadores, hacen que los uno o más procesadores realicen el procedimiento de cualquiera de las reivindicaciones 1 a 8.

15 15. Un dispositivo para la codificación de vídeo, el dispositivo que comprende:

una memoria configurada para almacenar datos de vídeo;

un codificador de vídeo que comprende uno o más procesadores configurados para:

20 determinar codificar un primer bloque en el modo de fusión, en el que, en el modo de fusión, se construye una lista de candidatos de fusión que comprende candidatos del vector de movimiento basándose en bloques ya codificados; determinar que una precisión del vector de movimiento para un primer bloque es una precisión de un píxel entero; construir una lista de candidatos de fusión para el primer bloque, para incluir candidatos del vector de movimiento basándose en bloques ya codificados, en el que la lista de candidatos de fusión comprende un candidato del vector de movimiento con precisión fraccionaria;

25 seleccionar el candidato del vector de movimiento con precisión fraccionaria;

redondear el candidato del vector de movimiento con precisión fraccionaria para determinar un vector de movimiento con precisión de un píxel entero para el primer bloque;

localizar un bloque de referencia para el primer bloque usando el vector de movimiento con precisión de un píxel entero; y

30 codificar el primer bloque basándose en el bloque de referencia.



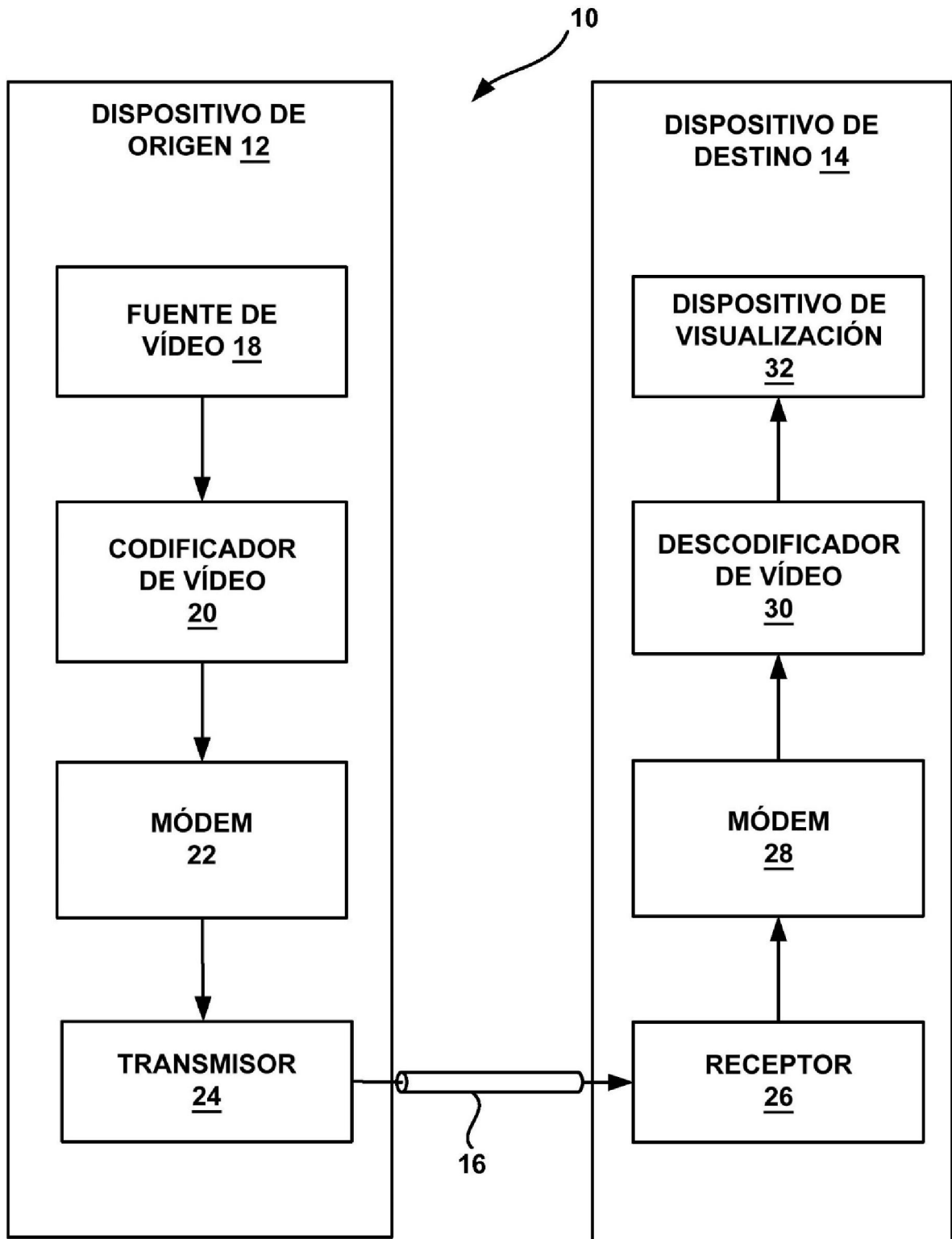


FIG. 1

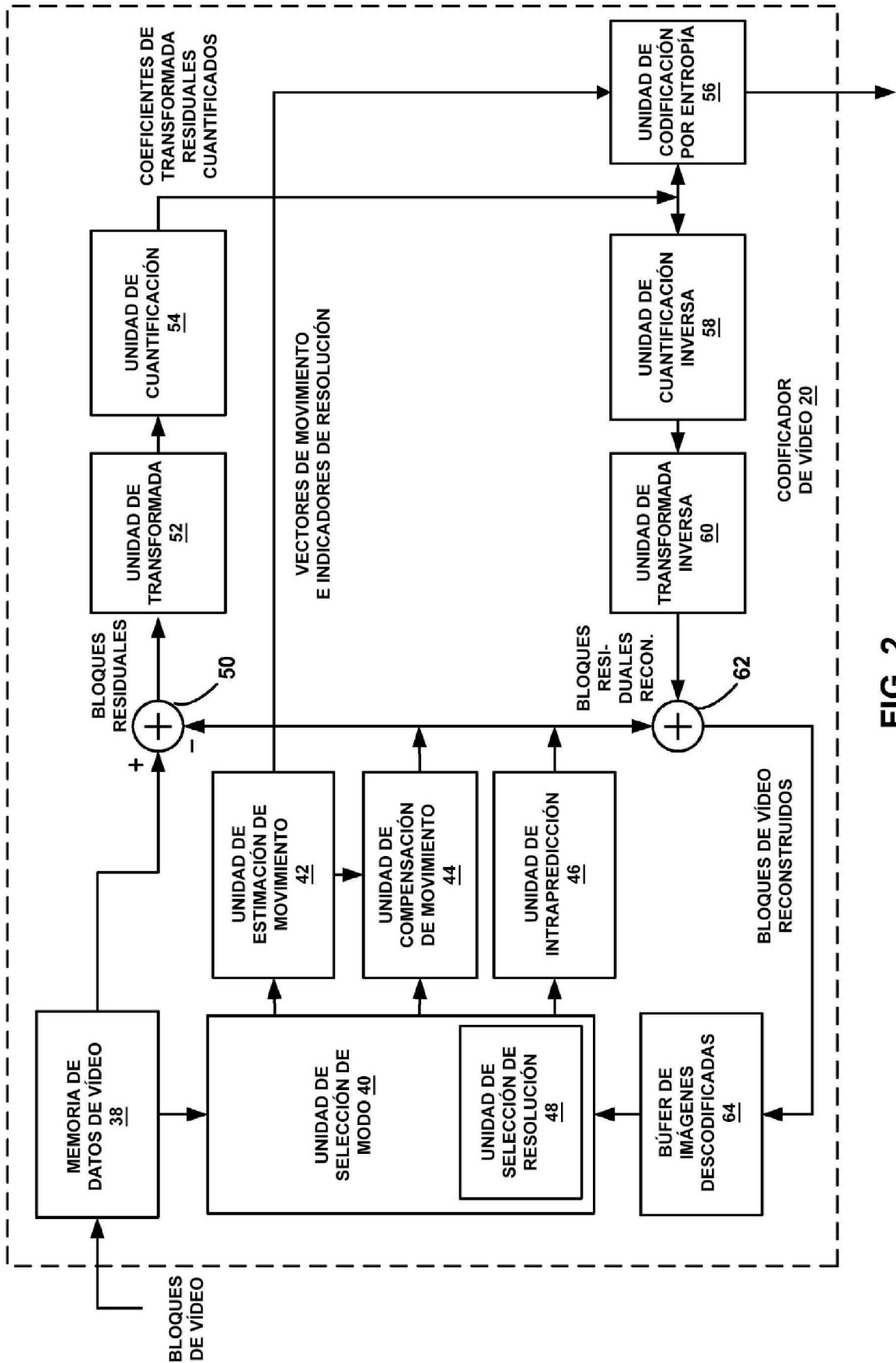


FIG. 2

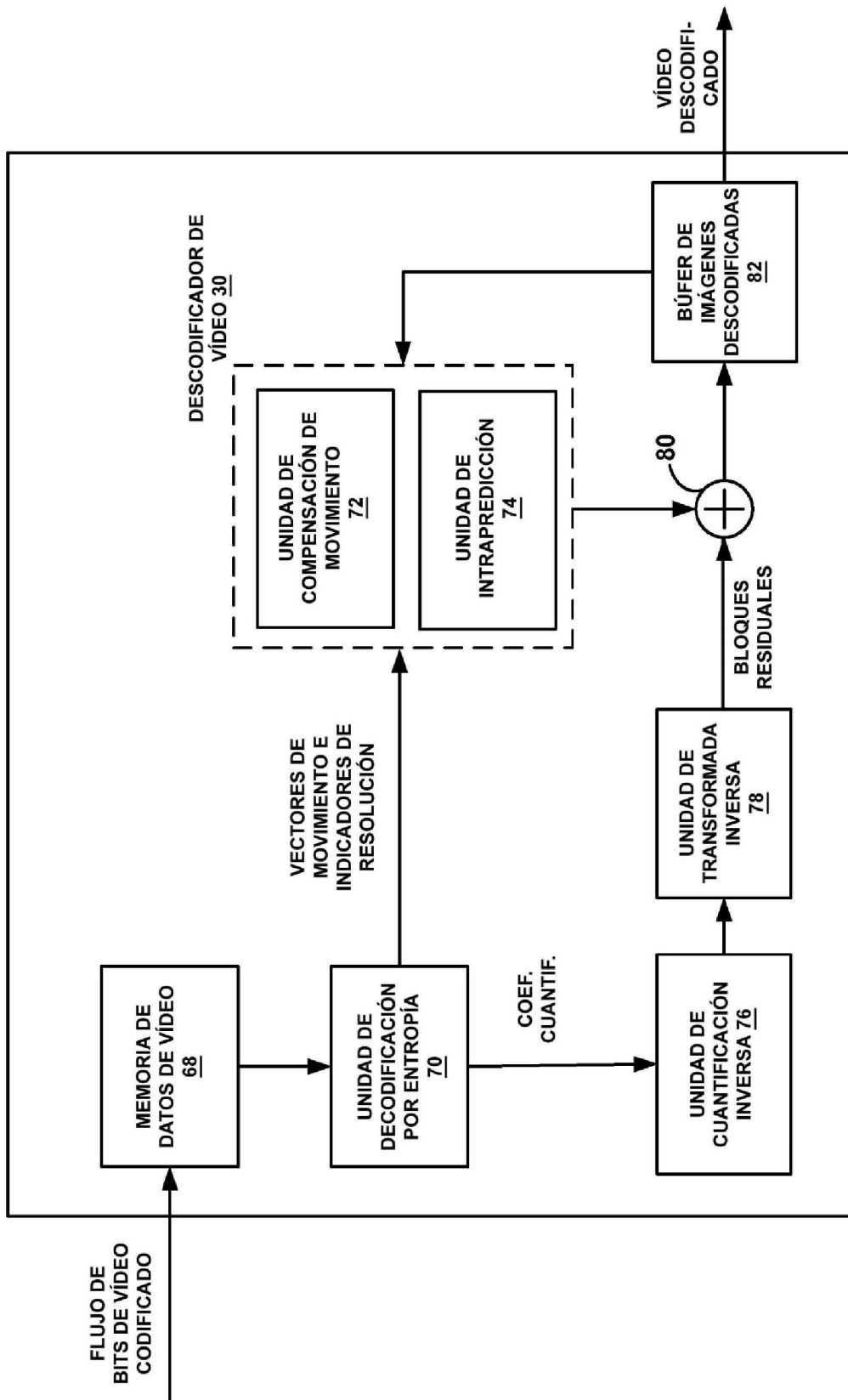
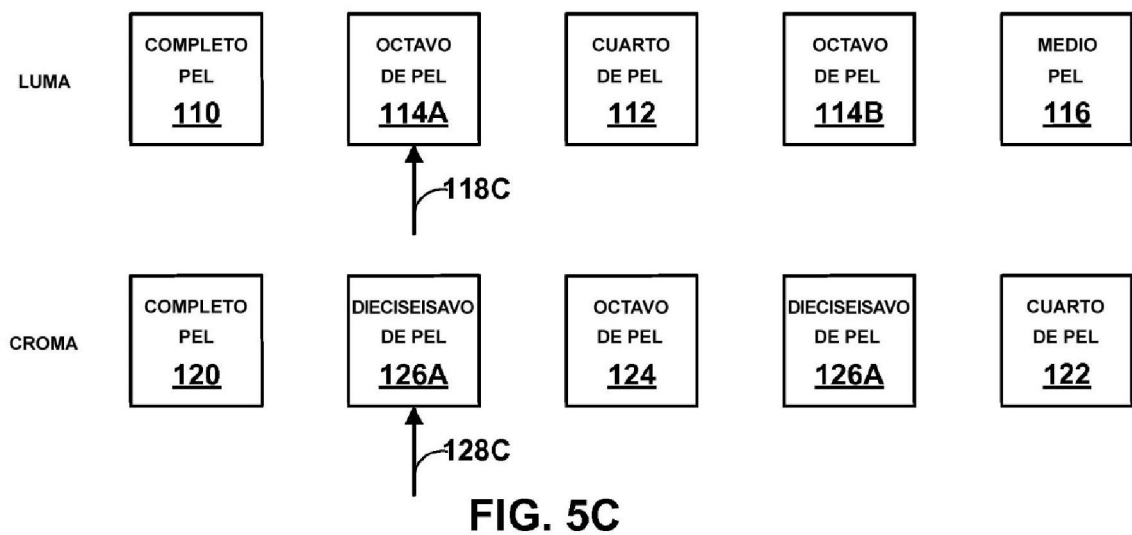
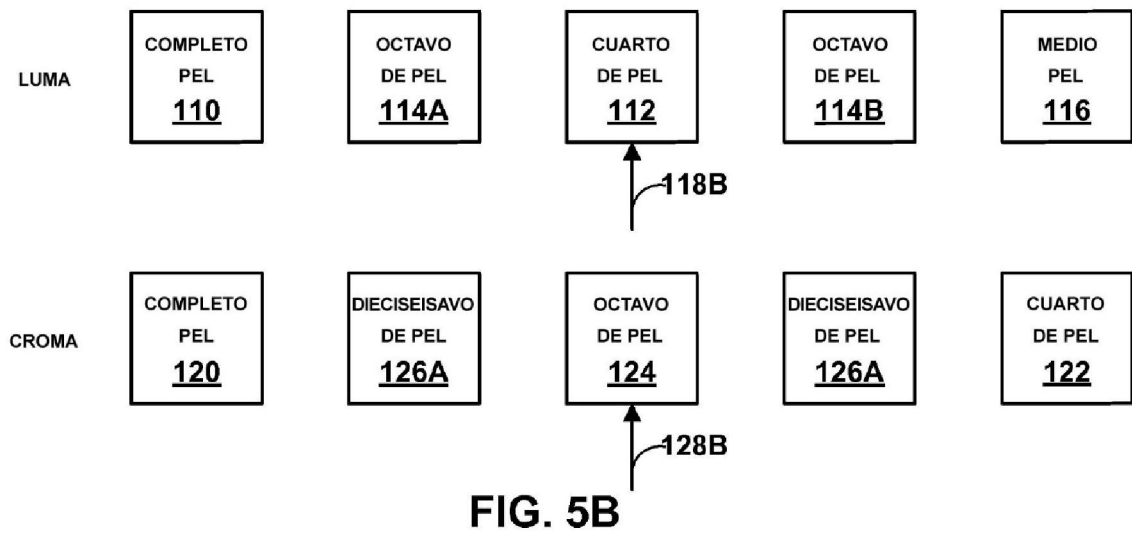
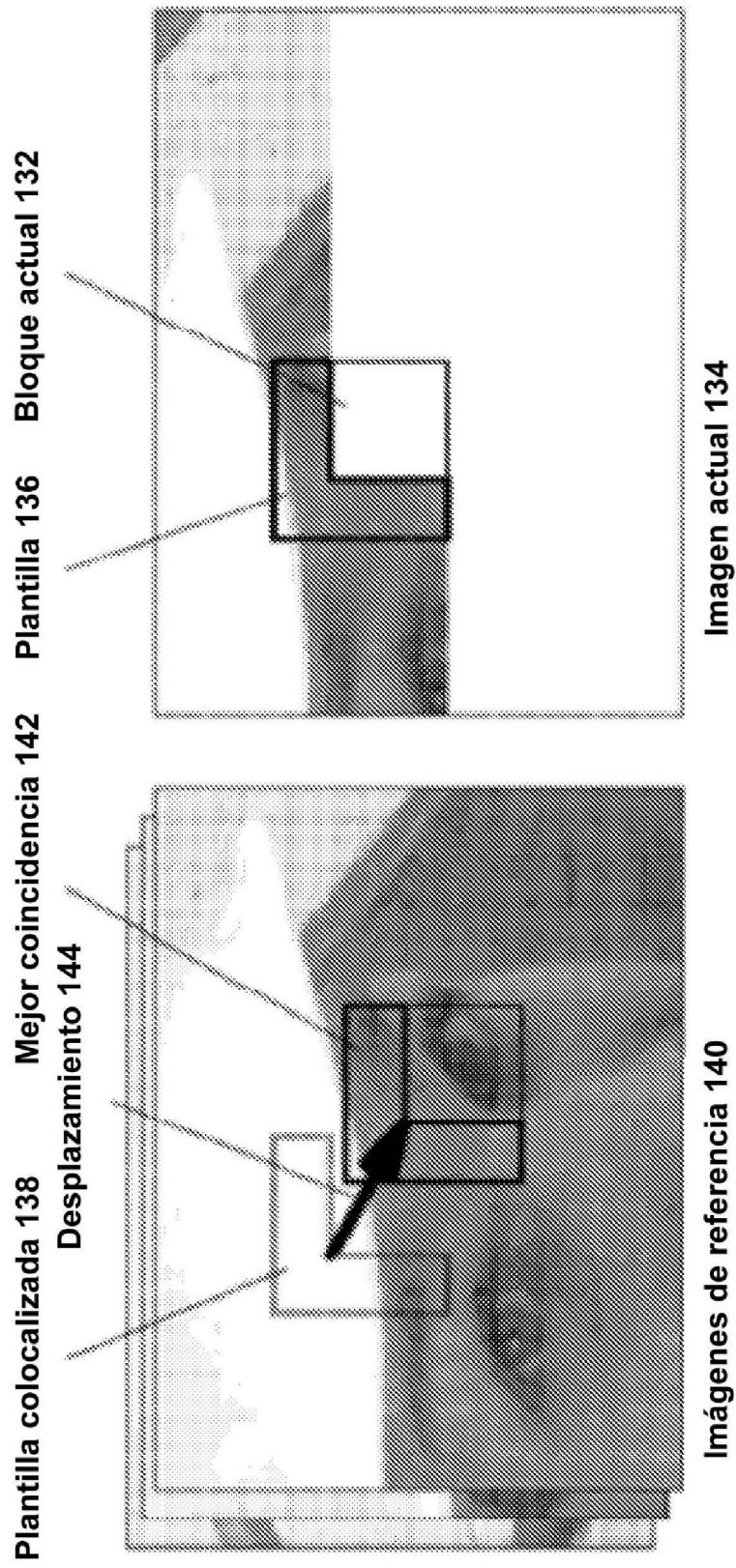


FIG. 3

COMPLETO DE PEL <b>100</b>	OCTAVO DE PEL <b>106A</b>	CUARTO DE PEL <b>104A</b>	OCTAVO DE PEL <b>106B</b>	MEDIO DE PEL <b>102A</b>	OCTAVO DE PEL <b>106C</b>	CUARTO DE PEL <b>104B</b>	OCTAVO DE PEL <b>106D</b>
OCTAVO DE PEL <b>106E</b>	OCTAVO DE PEL <b>106F</b>	OCTAVO DE PEL <b>106G</b>	OCTAVO DE PEL <b>106H</b>	OCTAVO DE PEL <b>106I</b>	OCTAVO DE PEL <b>106J</b>	OCTAVO DE PEL <b>106K</b>	OCTAVO DE PEL <b>106L</b>
CUARTO DE PEL <b>104C</b>	OCTAVO DE PEL <b>106M</b>	CUARTO DE PEL <b>104D</b>	OCTAVO DE PEL <b>106N</b>	CUARTO DE PEL <b>104E</b>	OCTAVO DE PEL <b>106O</b>	CUARTO DE PEL <b>104F</b>	OCTAVO DE PEL <b>106P</b>
OCTAVO DE PEL <b>106Q</b>	OCTAVO DE PEL <b>106R</b>	OCTAVO DE PEL <b>106S</b>	OCTAVO DE PEL <b>106T</b>	OCTAVO DE PEL <b>106U</b>	OCTAVO DE PEL <b>106V</b>	OCTAVO DE PEL <b>106W</b>	OCTAVO DE PEL <b>106X</b>
MEDIO DE PEL <b>102B</b>	OCTAVO DE PEL <b>106Y</b>	CUARTO DE PEL <b>104G</b>	OCTAVO DE PEL <b>106Z</b>	MEDIO DE PEL <b>102C</b>	OCTAVO DE PEL <b>106AA</b>	CUARTO DE PEL <b>104H</b>	OCTAVO DE PEL <b>106AB</b>
OCTAVO DE PEL <b>106AC</b>	OCTAVO DE PEL <b>106AD</b>	OCTAVO DE PEL <b>106AE</b>	OCTAVO DE PEL <b>106AF</b>	OCTAVO DE PEL <b>106AG</b>	OCTAVO DE PEL <b>106AH</b>	OCTAVO DE PEL <b>106AI</b>	OCTAVO DE PEL <b>106AJ</b>
CUARTO DE PEL <b>104I</b>	OCTAVO DE PEL <b>106AK</b>	CUARTO DE PEL <b>104J</b>	OCTAVO DE PEL <b>106AL</b>	CUARTO DE PEL <b>104K</b>	OCTAVO DE PEL <b>106AM</b>	CUARTO DE PEL <b>104L</b>	OCTAVO DE PEL <b>106AN</b>
OCTAVO DE PEL <b>106AO</b>	OCTAVO DE PEL <b>106AP</b>	OCTAVO DE PEL <b>106AQ</b>	OCTAVO DE PEL <b>106AR</b>	OCTAVO DE PEL <b>106AS</b>	OCTAVO DE PEL <b>106AT</b>	OCTAVO DE PEL <b>106AU</b>	OCTAVO DE PEL <b>106AV</b>

FIG. 4





**FIG. 6**

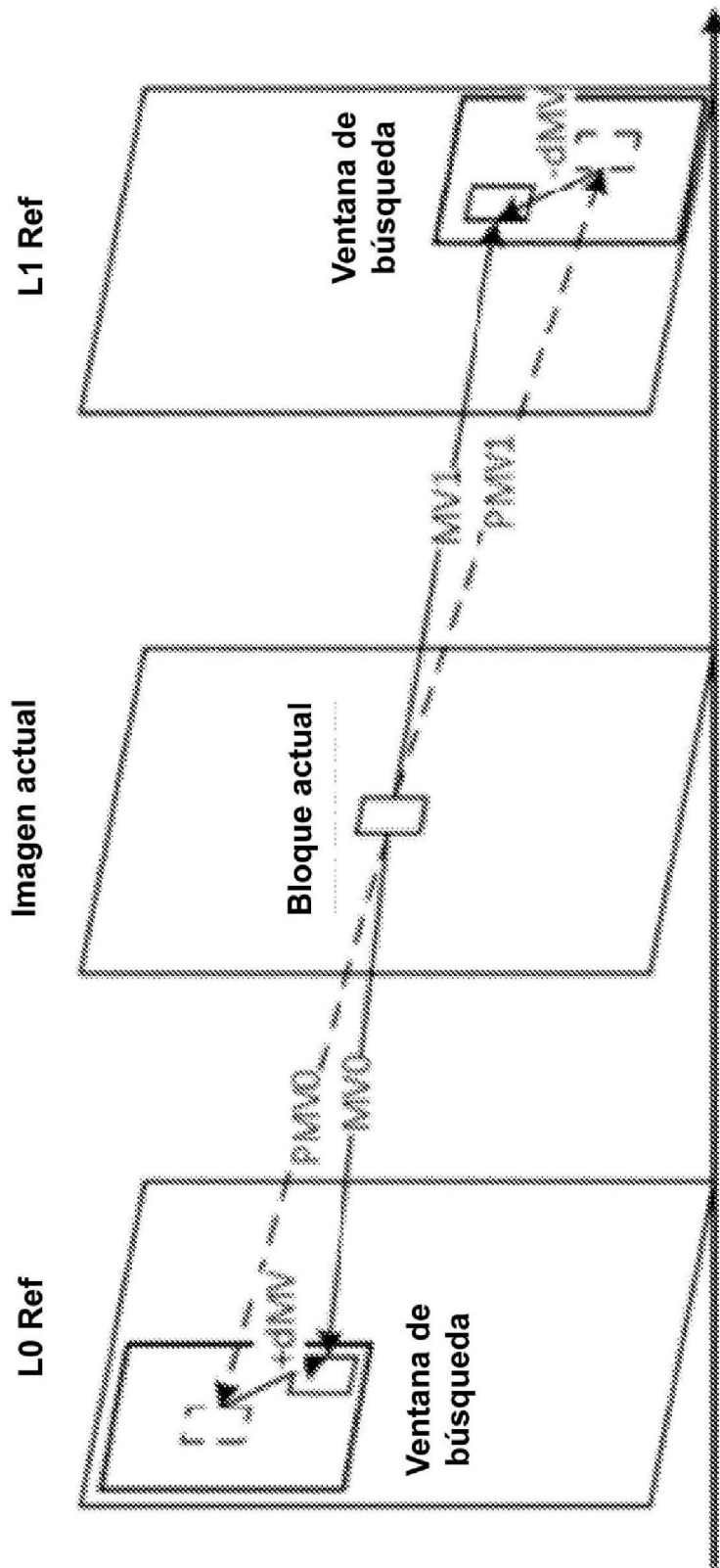


FIG. 7

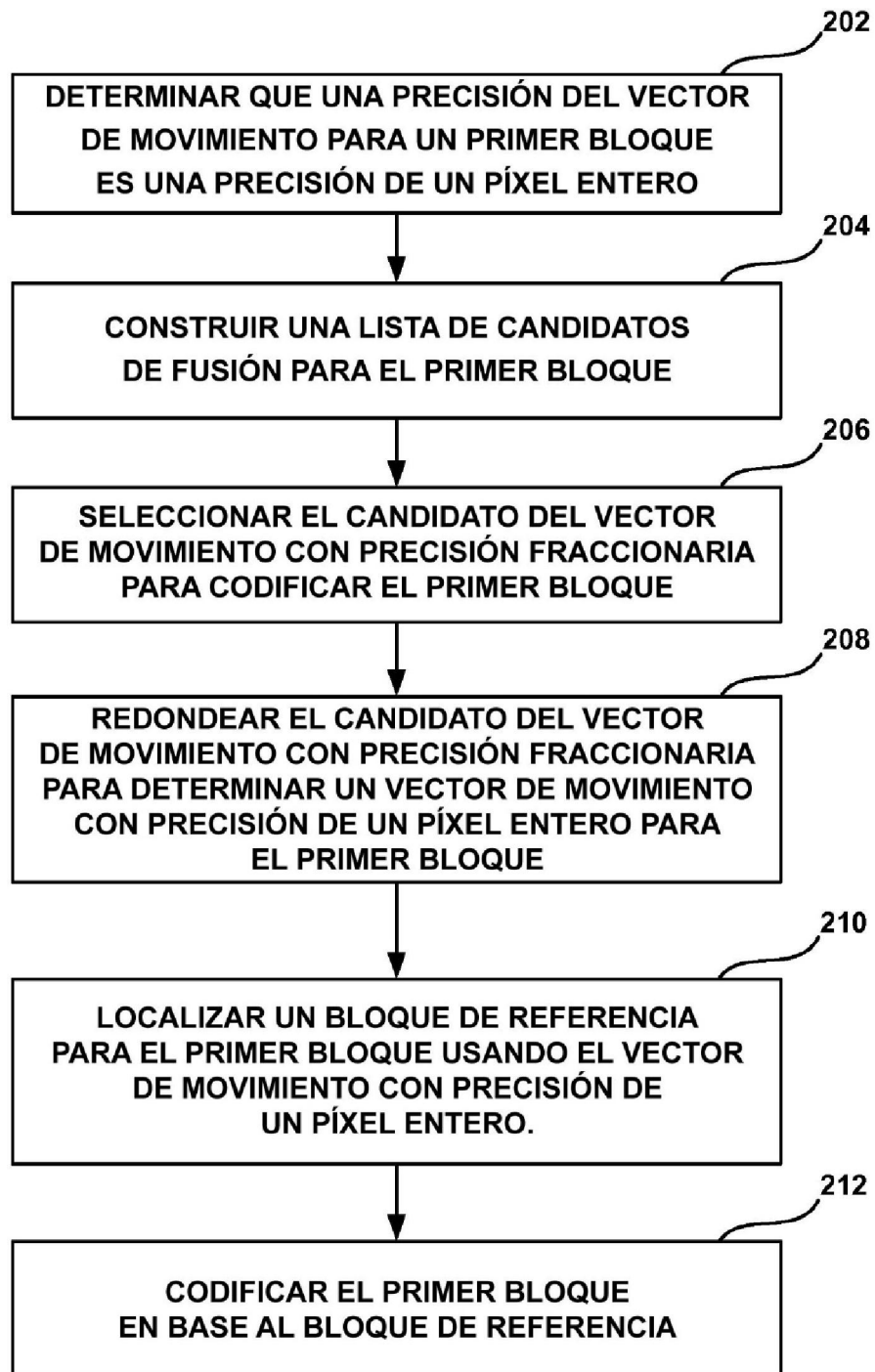


FIG. 8



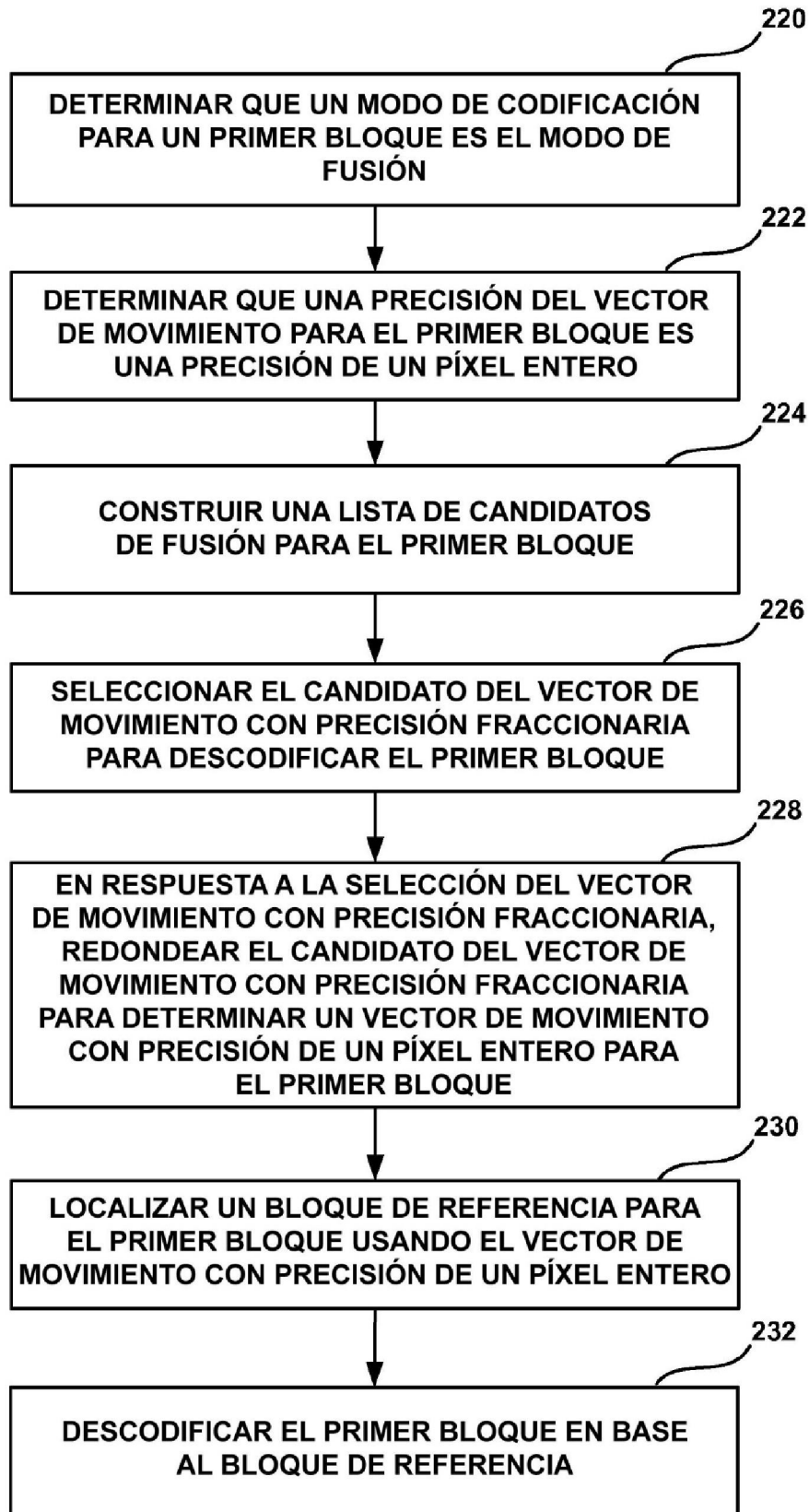


FIG. 9

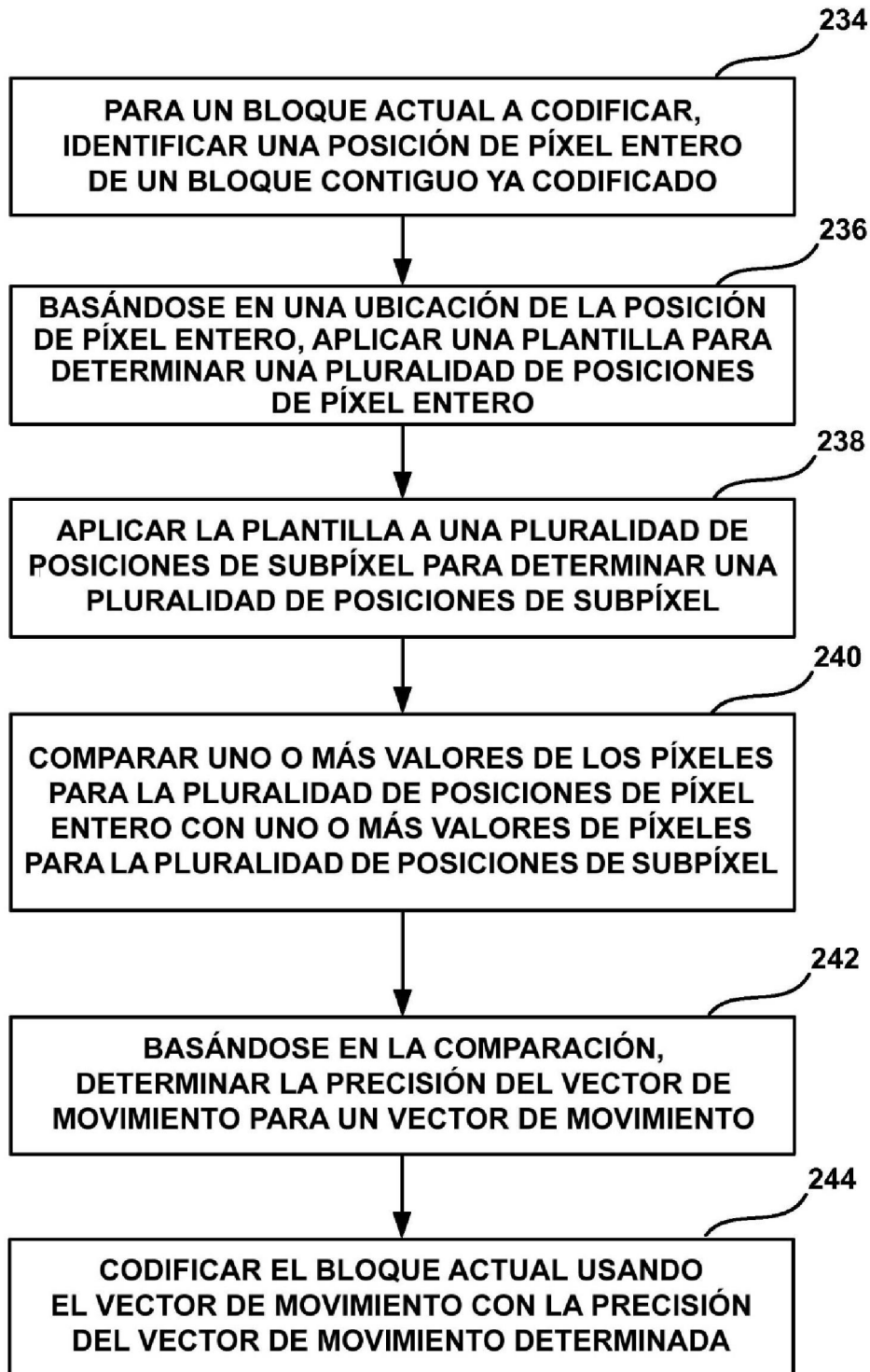
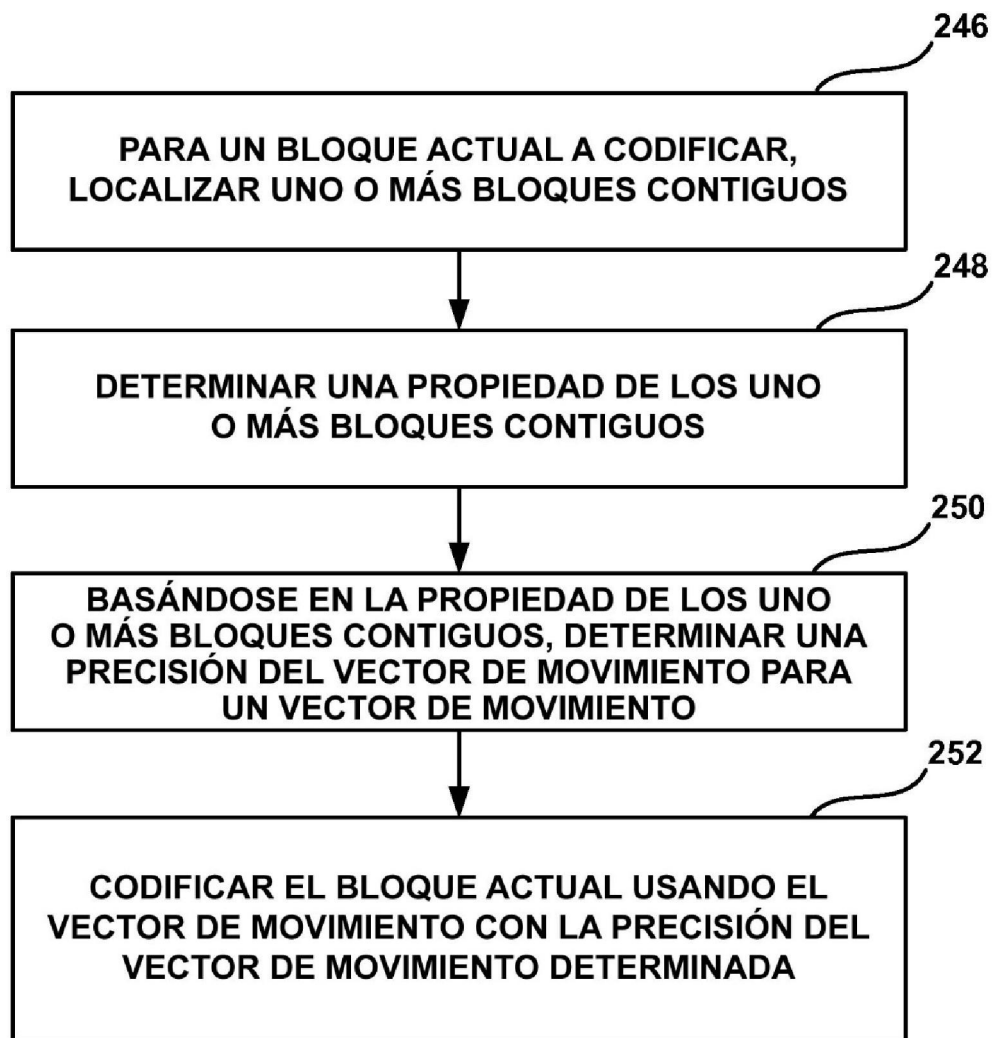


FIG. 10



**FIG. 11**

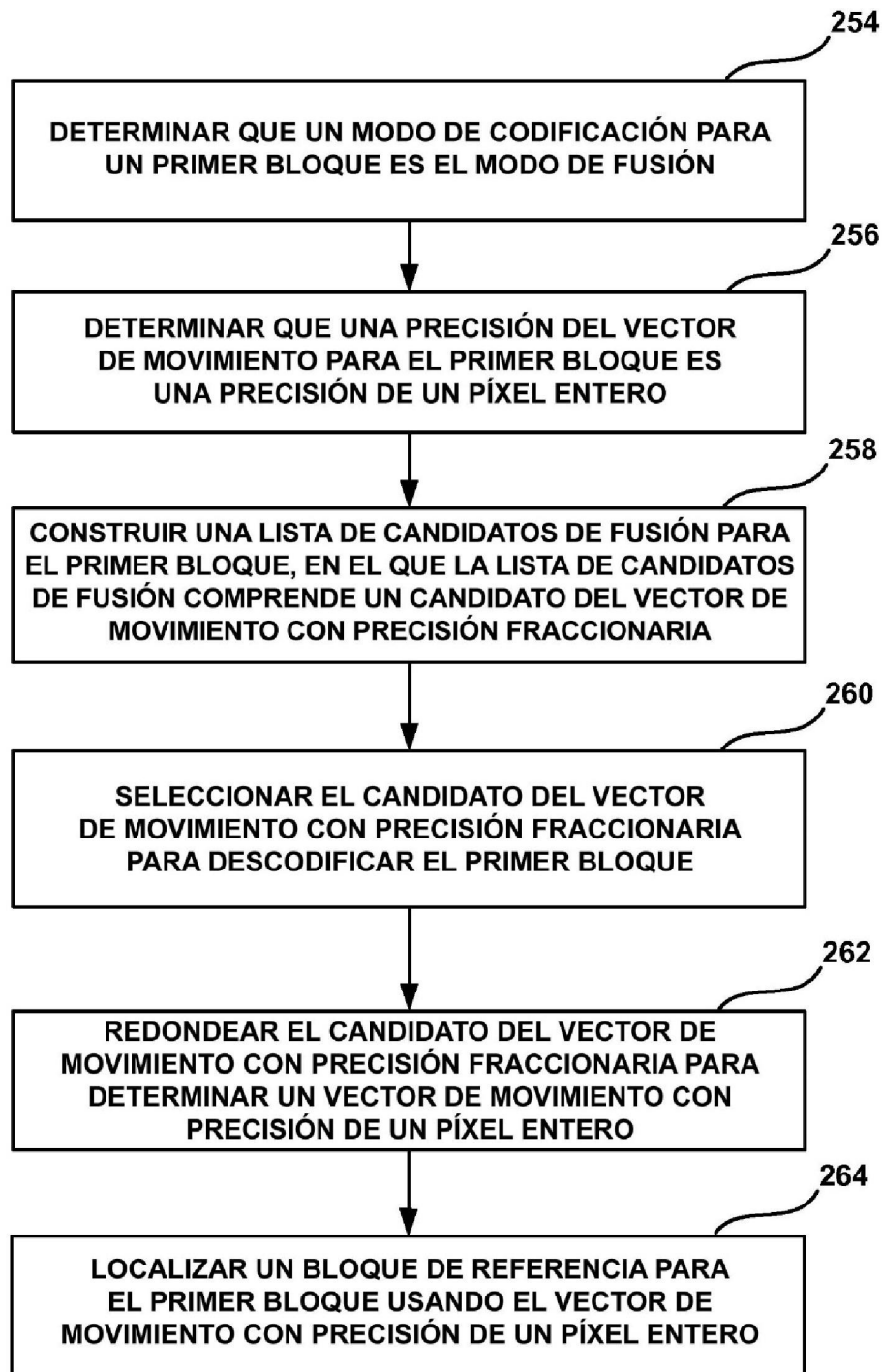
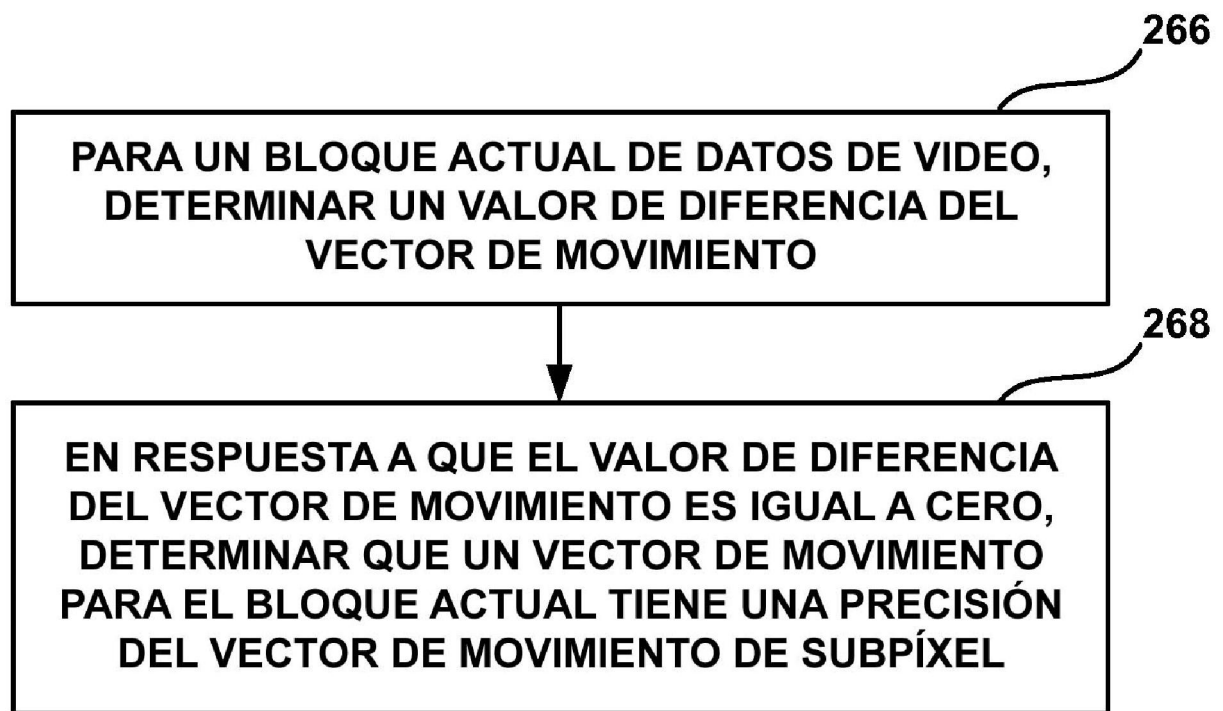
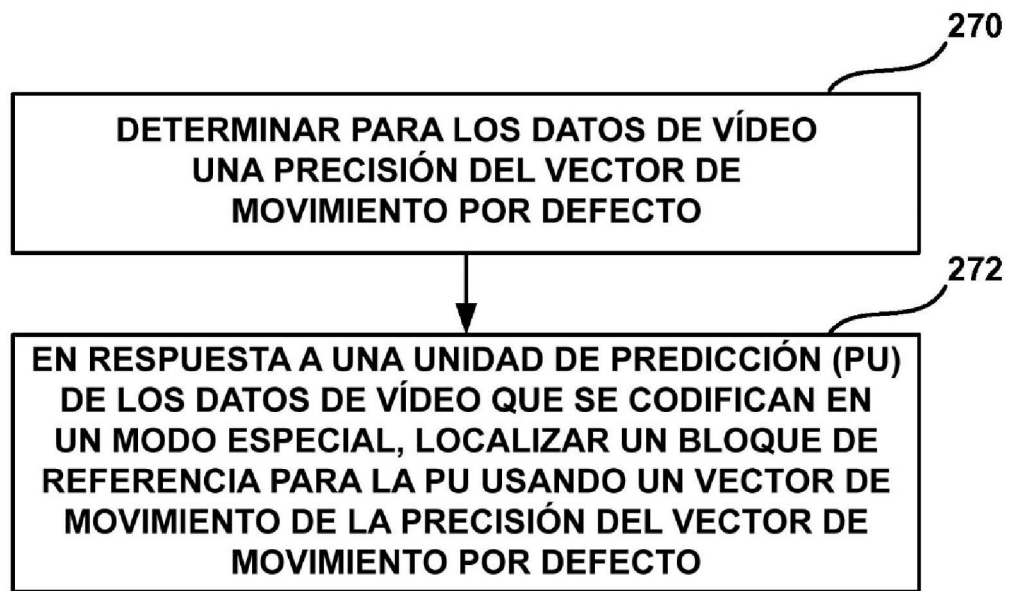


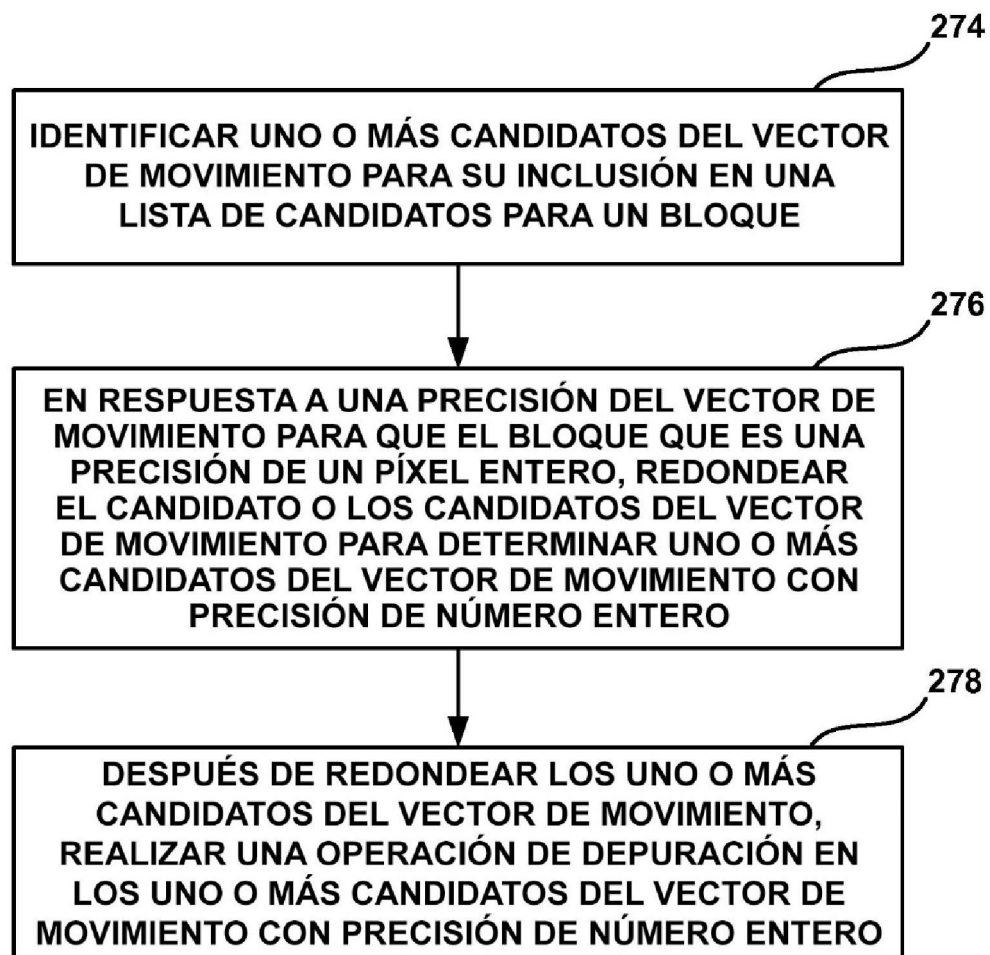
FIG. 12



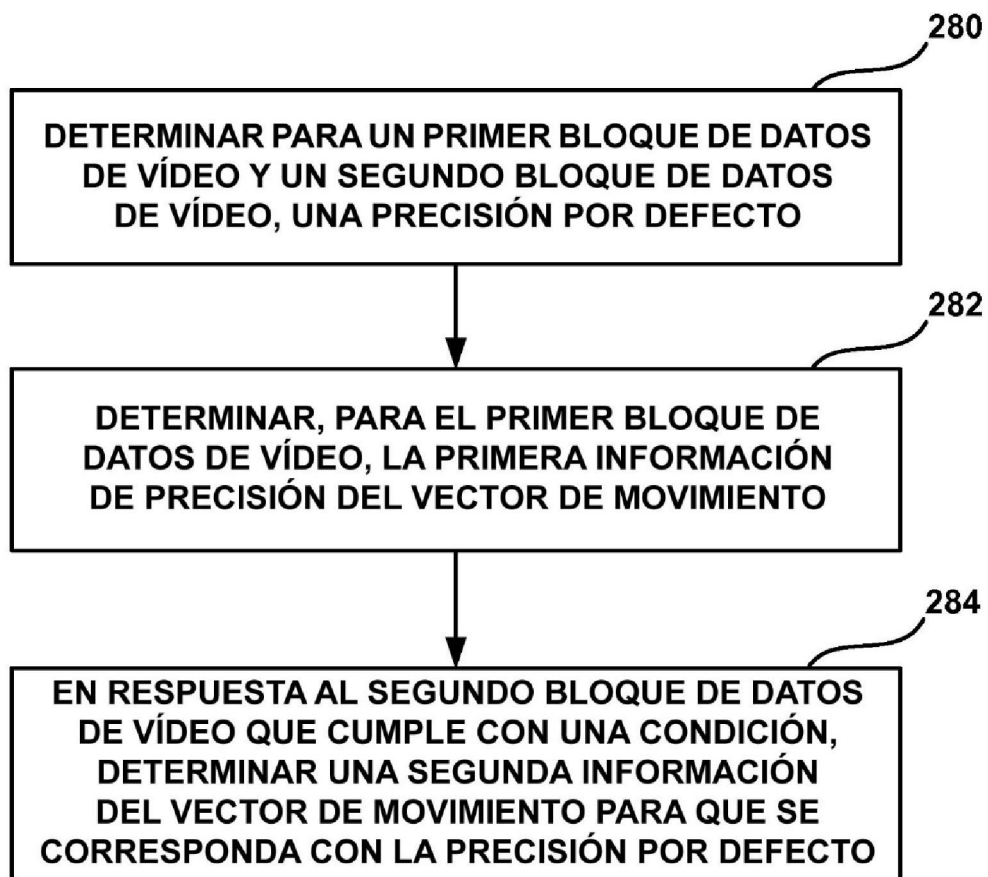
**FIG. 13**



**FIG. 14**



**FIG. 15**



**FIG. 16**



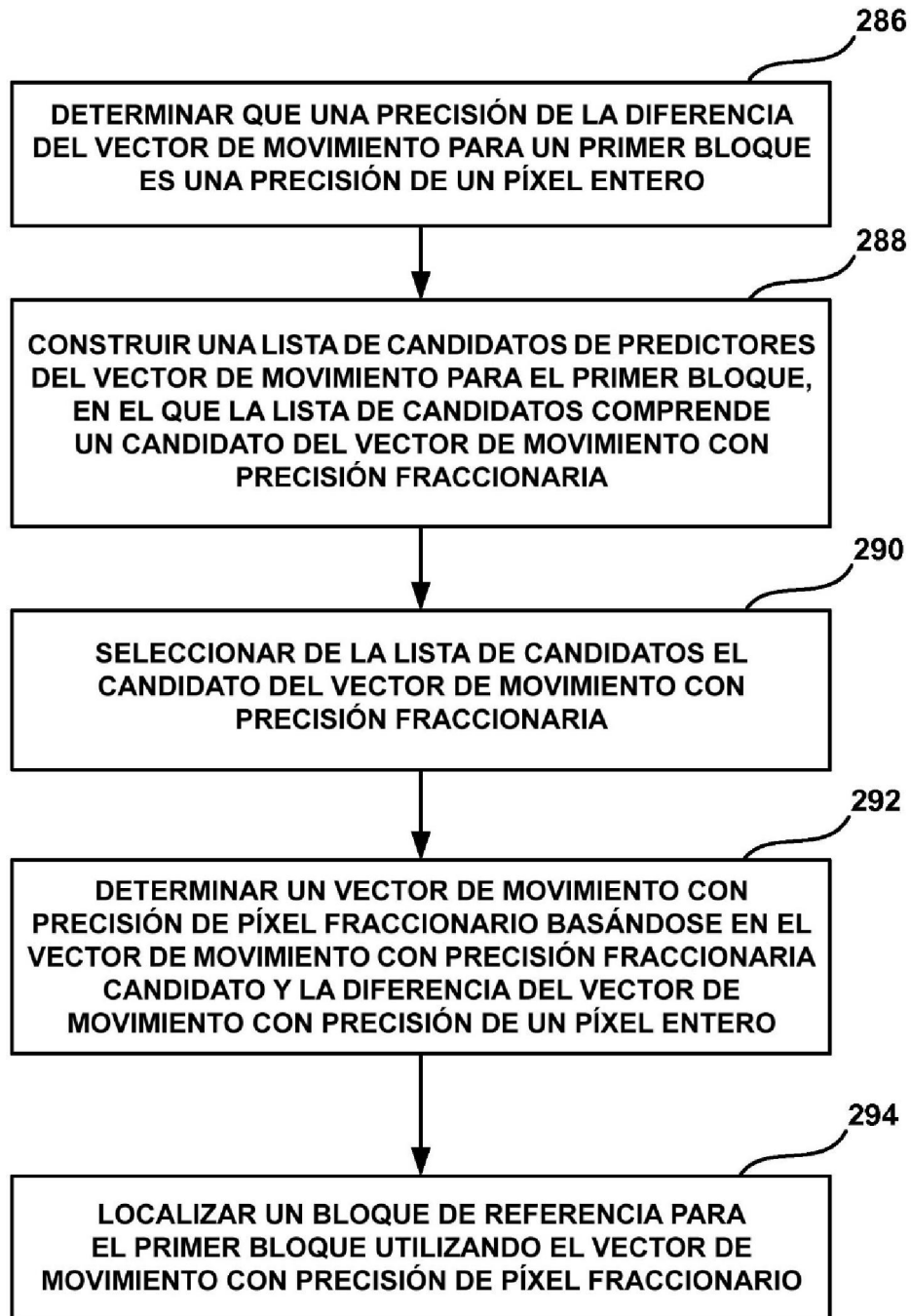


FIG. 17