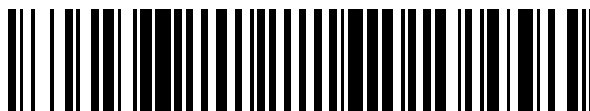


19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 900 029**

51 Int. Cl.:

**H04N 19/513** (2014.01)  
**H04N 19/46** (2014.01)  
**H04N 19/109** (2014.01)  
**H04N 19/176** (2014.01)  
**H04N 19/44** (2014.01)  
**H04N 19/56** (2014.01)  
**H04N 19/57** (2014.01)  
**H04N 19/96** (2014.01)  
**H04N 19/70** (2014.01)

12

## TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **25.03.2016** **PCT/US2016/024332**  
87 Fecha y número de publicación internacional: **06.10.2016** **WO16160608**  
96 Fecha de presentación y número de la solicitud europea: **25.03.2016** **E 16718538 (8)**  
97 Fecha y número de publicación de la concesión europea: **17.11.2021** **EP 3275188**

54 Título: **Derivación de vector de movimiento en codificación de vídeo**

30 Prioridad:

**27.03.2015 US 201562139572 P**  
**19.06.2015 US 201562182367 P**  
**24.03.2016 US 201615080478**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:  
**15.03.2022**

73 Titular/es:

**QUALCOMM INCORPORATED (100.0%)**  
**ATTN: International IP Administration 5775**  
**Morehouse Drive**  
**San Diego, California 92121-1714, US**

72 Inventor/es:

**LI, XIANG;**  
**CHEN, YING;**  
**ZHANG, LI;**  
**LIU, HONGBIN;**  
**CHEN, JIANLE y**  
**KARCZEWICZ, MARTA**

74 Agente/Representante:

**ISERN JARA, Jorge**

ES 2 900 029 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

Derivación de vector de movimiento en codificación de vídeo

## 5 CAMPO TÉCNICO

La presente divulgación se refiere a la codificación de vídeo.

## ANTECEDENTES

10

Las capacidades de vídeo digital se pueden incorporar a una amplia gama de dispositivos, que incluyen televisores digitales, sistemas de radiodifusión digital directa, sistemas de radiodifusión inalámbrica, asistentes personales digitales (PDA), ordenadores portátiles o de escritorio, ordenadores de tableta, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos móviles o de radio por satélite, los denominados "teléfonos inteligentes", dispositivos de videoconferencia, dispositivos de transmisión continua de vídeo y similares. Los dispositivos de vídeo digital implementan técnicas de codificación de vídeo, tales como las descritas en las normas definidas por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, parte 10, codificación de vídeo avanzada (AVC), la norma de codificación de vídeo de alta eficacia (HEVC) y las extensiones de dichas normas. Los dispositivos de vídeo pueden transmitir, recibir, codificar, descodificar y/o almacenar información de vídeo digital más eficazmente implementando dichas técnicas de codificación de vídeo.

15

20

25

30

Las técnicas de codificación de vídeo incluyen la predicción espacial (intraimagen) y/o la predicción temporal (interimagen) para reducir o eliminar la redundancia intrínseca en las secuencias de vídeo. Para la codificación de vídeo basada en bloques, un fragmento de vídeo (por ejemplo, una trama de vídeo o una parte de una trama de vídeo) se puede dividir en bloques de vídeo, que para algunas técnicas también se pueden denominar bloques de árbol, unidades de codificación (CU) y/o nodos de codificación. Los bloques de vídeo en un fragmento intracodificado (I) de una imagen se codifican usando predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen. Los bloques de vídeo de un fragmento intercodificado (P o B) de una imagen pueden usar predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen o predicción temporal con respecto a muestras de referencia en otras imágenes de referencia. Las imágenes se pueden denominar tramas, y las imágenes de referencia se pueden denominar tramas de referencia.

35

40

La predicción espacial o temporal da como resultado un bloque predictivo para un bloque que se va a codificar. Los datos residuales representan diferencias de píxeles entre el bloque original que se va a codificar y el bloque predictivo. Un bloque intercodificado se codifica de acuerdo con un vector de movimiento que apunta a un bloque de muestras de referencia que forman el bloque predictivo, e indicando los datos residuales la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intracodificado se codifica de acuerdo con un modo de intracodificación y los datos residuales. Para compresión adicional, los datos residuales se pueden transformar desde el dominio de píxel a un dominio de transformada, dando como resultado coeficientes de transformada residuales que, a continuación, se pueden cuantificar. Se pueden explorar los coeficientes de transformada cuantificados, dispuestos inicialmente en una matriz bidimensional, para producir un vector unidimensional de coeficientes de transformada, y se puede aplicar codificación entrópica para lograr incluso más compresión. Los siguientes documentos divulgan dichas técnicas de codificación de vídeo:

45

El documento US 2011/176611 A1 (HUANG YU-WEN [TW] *ET AL.*) 21 de julio de 2011 (21-07-2011)

50

Y-J CHIU *ET AL.*: "CE1 Subtest1: A joint proposal of candidate-based decoder-side motion vector derivation", 95. REUNIÓN DE MPEG; 24-1-2011 - 28-1-2011; DAEGU; (GRUPO DE EXPERTOS EN IMÁGENES EN MOVIMIENTO O ISO/IEC JTC1/SC29/WG11), n.º m19473, 26 de enero de 2011 (26-01-2011)

55

STEFFEN KAMP *ET AL.*: "Decoder Side Motion Vector Derivation", 82. REUNIÓN DE MPEG; 22-10-2007 - 26-10-2007; SHENZHEN; (GRUPO DE EXPERTOS EN IMÁGENES EN MOVIMIENTO O ISO/IEC JTC1/SC29/WG11), n.º M14917, 16 de octubre de 2007 (16-10-2007)

60

ZHAO X *ET AL.*: "CE3 related: Simplifications to sub-PU level inter-view motion prediction", 6. REUNIÓN DE JCT-3V; 25-10-2013 - 1-11-2013; GINEBRA; (EQUIPO DE COLABORACIÓN CONJUNTA EN DESARROLLO DE EXTENSIONES DE CODIFICACIÓN DE VÍDEO EN 3D DE ISO/IEC JTC1/SC29/WG11 E ITU-T SG.16); URL: [HTTP://PHENIX.INT-EVRY.FR/JCT2/](http://phenix.int-evry.fr/JCT2/), n.º JCT3V-F0127, 18 de octubre de 2013 (18-10-2013)

65

LIN S *ET AL.*: "TE1: Huawei report on DMVD improvements", 2. REUNIÓN DE JCT-VC; 21-7-2010 - 28-7-2010; GINEBRA; (EQUIPO DE COLABORACIÓN CONJUNTA EN CODIFICACIÓN DE VÍDEO DE ISO/IEC JTC1/SC29/WG11 E ITU-T SG.16); URL: [HTTP://WFTP3.ITU.INT/AV-ARCH/JCTVC-SITE/](http://wftp3.itu.int/AV-ARCH/JCTVC-SITE/), n.º JCTVC-B037, 23 de julio de 2010 (23-07-2010)

Y-J CHIU (INTEL) *ET AL.*: "Video coding technology proposal by Intel", 1. REUNIÓN DE JCT-VC; 15-4-2010 - 23-4-2010; DRESDE; (EQUIPO DE COLABORACIÓN CONJUNTA EN CODIFICACIÓN DE VÍDEO DE ISO/IEC JTC1/SC29/WG11 E ITU-TSG.16); URL: [HTTP://WFTP3.ITU.INT/AV-ARCH/JCTVC-SITE/](http://WFTP3.ITU.INT/AV-ARCH/JCTVC-SITE/), n.º JCTVC-A106, 18 de abril de 2010 (18-04-2010)

HM7: MODELO DE PRUEBA DE CODIFICACIÓN DE VÍDEO DE ALTA EFICACIA (HEVC) 7 DESCRIPCIÓN DEL CODIFICADOR: "High Efficiency Video Coding (HEVC) Test Model 7 (HM 7) Encoder Description", 9. REUNIÓN DE JCT-VC; 100. REUNIÓN DE JCT-VC; 27-4-2012 - 7-5-2012; GINEBRA; (EQUIPO DE COLABORACIÓN CONJUNTA EN CODIFICACIÓN DE VÍDEO DE ISO/IEC JTC1/SC29/WG11 E ITU-T SG.16); URL: [HTTP://WFTP3.ITU.INT/AV-ARCH/JCTVC-SITE/](http://WFTP3.ITU.INT/AV-ARCH/JCTVC-SITE/), n.º JCTVC-I1002, 9 de julio de 2012 (09-07-2012)

TANG K L *ET AL.*: "Enhanced SAD reuse fast motion estimation", COMUNICACIONES VISUALES Y PROCESAMIENTO DE IMÁGENES; 30-1-2007 - 1-2-2007; SAN JOSÉ, 30 de enero de 2007 (30-01-2007)

## SUMARIO

Las técnicas de la presente divulgación se refieren a derivar información de movimiento en la codificación de vídeo. Por ejemplo, un codificador de vídeo (un codificador de vídeo o decodificador de vídeo) puede generar información de movimiento que no está incluida en el flujo de bits para codificar datos de vídeo que están incluidos en el flujo de bits. Derivando la información de movimiento en el bucle de codificación o decodificación, se puede lograr un ahorro de bits en relación con las técnicas que incluyen información de movimiento en el flujo de bits, tales como las técnicas de interpredicción tradicionales.

El alcance de protección buscado por la invención de la presente divulgación se define de acuerdo con las reivindicaciones adjuntas.

Los detalles de uno o más ejemplos de la divulgación se exponen en los dibujos adjuntos y en la descripción siguiente. Otros rasgos característicos, objetivos y ventajas resultarán evidentes a partir de la descripción, los dibujos y 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 implementar las técnicas de la presente divulgación.

La FIG. 2 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo que puede implementar las técnicas de la presente divulgación.

La FIG. 3 es un diagrama de bloques que ilustra un ejemplo de decodificador de vídeo que puede implementar las técnicas de la presente divulgación.

Las FIGS. 4A y 4B son diagramas conceptuales que ilustran candidatos de vectores de movimiento vecinos espaciales de ejemplo para un modo de fusión y un modo de predicción de vector de movimiento avanzada (AMVP).

Las FIGS. 5A y 5B son diagramas conceptuales que ilustran un candidato de predictor de vector de movimiento temporal (TMVP) de ejemplo y un escalado de vector de movimiento.

La FIG. 6 es un diagrama conceptual que ilustra un ejemplo de estimación de movimiento (ME) unilateral en conversión ascendente de velocidad de tramas (FRUC).

La FIG. 7 es un diagrama conceptual que ilustra un ejemplo de estimación de movimiento (ME) bilateral en FRUC.

La FIG. 8 es un diagrama conceptual que ilustra un ejemplo de derivación de vector de movimiento de lado de decodificador (DMVD) basado en coincidencia de plantillas.

La FIG. 9 es un diagrama conceptual que ilustra un ejemplo de derivación de vector de movimiento bidireccional con base especular en DMVD

La FIG. 10 es un diagrama conceptual que ilustra la derivación de vector de movimiento basada en coincidencia bilateral extendida.

La FIG. 11 es un diagrama de flujo que ilustra un ejemplo de decodificación de una unidad de predicción (PU) usando DMVD

La FIG. 12 es un diagrama de flujo que ilustra un proceso de ejemplo para determinar un modo de derivación de información de movimiento para codificar un bloque de datos de vídeo.

La FIG. 13 es un diagrama de flujo que ilustra un proceso de ejemplo para derivar un vector de movimiento para codificar un bloque de datos de vídeo.

La FIG. 14 es un diagrama de flujo que ilustra un proceso de ejemplo para derivar información de movimiento para subbloques de un bloque de datos de vídeo.

## DESCRIPCIÓN DETALLADA

Las técnicas de la presente divulgación se refieren a la derivación de información de movimiento de lado de decodificador, la división de bloques y/o la interpolación de datos de vídeo en la codificación de vídeo basada en

bloques. Las técnicas se pueden aplicar a cualquiera de los códecs de vídeo existentes, tales como la codificación de vídeo de alta eficacia (HEVC) o ser una herramienta de codificación eficaz para cualquier norma de codificación de vídeo futura.

Los dispositivos de codificación de vídeo implementan técnicas de compresión de vídeo para codificar y descodificar datos de vídeo eficazmente. Las técnicas de compresión de vídeo pueden incluir la aplicación de predicción espacial (por ejemplo, predicción intratrama), predicción temporal (por ejemplo, predicción entre tramas) y/u otras técnicas de predicción para reducir o eliminar la redundancia intrínseca en las secuencias de vídeo. Un codificador de vídeo típicamente divide cada imagen de una secuencia de vídeo original en regiones rectangulares denominadas bloques de vídeo o unidades de codificación (descritas con más detalle a continuación). Estos bloques de vídeo se pueden codificar usando un modo de predicción particular.

Para los modos de interpredicción, un codificador de vídeo busca típicamente un bloque similar al que se codifica en una trama en otra ubicación temporal, denominada trama de referencia. El codificador de vídeo puede restringir la búsqueda a un determinado desplazamiento espacial desde el bloque que se va a codificar. Se puede ubicar la mejor coincidencia usando un vector de movimiento bidimensional (2D) que incluye un componente de desplazamiento horizontal y un componente de desplazamiento vertical. Para un modo de intrapredicción, un codificador de vídeo puede formar el bloque predicho usando técnicas de predicción espacial basadas en datos de bloques vecinos codificados previamente dentro de la misma imagen.

El codificador de vídeo puede determinar un error de predicción, es decir, la diferencia entre los valores de píxeles en el bloque que se codifica y el bloque predicho (también denominado residual). El codificador de vídeo también puede aplicar una transformada al error de predicción, tal como una transformada de coseno discreta (DCT), para generar coeficientes de transformada. Después de la transformación, el codificador de vídeo puede cuantificar los coeficientes de transformada. Los coeficientes de transformada cuantificados y los vectores de movimiento se pueden representar usando elementos de sintaxis y, junto con la información de control, formar una representación codificada de una secuencia de vídeo. En algunos casos, el codificador de vídeo puede entropizar elementos de sintaxis de código, reduciendo además de este modo el número de bits necesarios para su representación.

Un descodificador de vídeo puede, usando los elementos de sintaxis y la información de control analizados anteriormente, construir datos predictivos (por ejemplo, un bloque predictivo) para descodificar una trama actual. Por ejemplo, el descodificador de vídeo puede añadir el bloque predicho y el error de predicción comprimido. El descodificador de vídeo puede determinar el error de predicción comprimido ponderando las funciones de base de transformada usando los coeficientes cuantificados. La diferencia entre la trama reconstruida y la trama original se llama error de reconstrucción.

En algunos casos, un descodificador de vídeo o un dispositivo de posprocesamiento pueden interpolar imágenes en base a una o más imágenes de referencia. Dichas imágenes interpoladas no se incluyen en un flujo de bits codificado. El descodificador de vídeo o el dispositivo de posprocesamiento pueden interpolar imágenes para convertir de forma ascendente una velocidad de trama original de un flujo de bits codificado. Este proceso se puede denominar conversión ascendente de velocidad de tramas (FRUC). De forma alternativa, el descodificador de vídeo o el dispositivo de posprocesamiento pueden interpolar imágenes para insertar una o más imágenes que se omitieron por un codificador de vídeo para codificar una secuencia de vídeo a una velocidad de trama reducida. En cualquier caso, el descodificador de vídeo o el dispositivo de posprocesamiento interpola las tramas que no están incluidas en un flujo de bits codificado que se ha recibido por el descodificador de vídeo. El descodificador de vídeo o el dispositivo de posprocesamiento puede interpolar las imágenes usando cualquiera de una serie de técnicas de interpolación, por ejemplo, usando interpolación de tramas compensadas por movimiento, repetición de tramas o promediado de tramas.

Si bien se han usado determinadas técnicas para interpolar imágenes con el propósito de conversión ascendente, dichas técnicas no se han usado ampliamente durante la codificación de vídeo, por ejemplo, para codificar datos de vídeo que están incluidos en un flujo de bits codificado. Por ejemplo, las técnicas para interpolar imágenes pueden requerir relativamente mucho tiempo y/o requerir una cantidad relativamente grande de potencia de procesamiento. En consecuencia, dichas técnicas no se han realizado típicamente en bucle cuando se descodifican datos de vídeo.

De acuerdo con aspectos de la presente divulgación, un codificador de vídeo (por ejemplo, un codificador de vídeo o un descodificador de vídeo) puede derivar información de movimiento cuando codifica un flujo de bits. Por ejemplo, el codificador de vídeo puede generar información de movimiento que no está incluida en el flujo de bits para codificar datos de vídeo que están incluidos en el flujo de bits. Derivando la información de movimiento en el bucle de codificación o descodificación, se puede lograr un ahorro de bits en relación con las técnicas que incluyen información de movimiento en el flujo de bits (tales como las técnicas de interpredicción mencionadas anteriormente).

De acuerdo con algunos aspectos de la presente divulgación, un codificador de vídeo puede utilizar una pluralidad de técnicas de derivación de información de movimiento durante la codificación. En dichos ejemplos, el codificador

de vídeo puede determinar un modo de derivación de información de movimiento para determinar qué técnicas de derivación de información de movimiento usar cuando se determina la información de movimiento para un bloque actual. En general, usar un modo de derivación de información de movimiento para derivar información de movimiento puede incluir realizar una búsqueda de movimiento para un primer conjunto de datos de referencia que corresponde a un segundo conjunto de datos de referencia fuera del bloque actual. Por ejemplo, usando el modo de derivación de información de movimiento (por ejemplo, una técnica de coincidencia bilateral, una técnica de coincidencia de plantillas u otra técnica, como se describe con más detalle a continuación), el codificador de vídeo puede seleccionar un candidato de vector de movimiento en una lista de candidatos de vector de movimiento. El codificador de vídeo puede seleccionar el candidato de vector de movimiento en base al candidato de vector de movimiento que identifica los datos de referencia en una imagen de referencia que coincide relativamente de forma estrecha con los datos de la imagen actual (lo que se puede denominar determinar la "mejor coincidencia" de los datos de referencia).

En algunos casos, el codificador de vídeo puede usar el candidato de vector de movimiento seleccionado para identificar una ventana de búsqueda en una imagen de referencia. El codificador de vídeo puede refinar el candidato de vector de movimiento en base a datos de referencia en la ventana de búsqueda que coinciden relativamente de forma estrecha con los datos correspondientes en la imagen actual. Es decir, el codificador de vídeo puede derivar nueva información de movimiento para el bloque actual en base al movimiento entre los datos de referencia en la ventana de búsqueda que coinciden de forma estrecha con los datos en la imagen actual. A continuación, el codificador de vídeo puede realizar una compensación de movimiento para el bloque actual usando la información de movimiento derivada. De esta manera, el codificador de vídeo puede derivar información de movimiento para un bloque actual sin que se señalice la información de movimiento en un flujo de bits codificado.

De acuerdo con aspectos de la presente divulgación, en algunos ejemplos, el codificador de vídeo puede dividir un bloque de datos de vídeo en subbloques cuando deriva información de movimiento. Por ejemplo, el codificador de vídeo puede derivar por separado información de movimiento para cada subbloque de un bloque más grande. En algunos casos, el codificador de vídeo puede determinar inicialmente la información de movimiento para el bloque y usar la información de movimiento derivada como información de movimiento candidata para cada uno de los subbloques. A continuación, el codificador de vídeo puede refinar además la información de movimiento derivada para cada uno de los subbloques, por ejemplo, usando un modo de derivación de información de movimiento (por ejemplo, una técnica de coincidencia bilateral, una técnica de coincidencia de plantillas u otra técnica, como se describe con más detalle a continuación).

Las técnicas de la presente divulgación también pueden incluir técnicas para interpolar imágenes. En algunos casos, se puede usar cualquier combinación de las técnicas anteriores para interpolar una imagen que no está incluida en el flujo de bits, por ejemplo, similar a la conversión ascendente de velocidad de tramas. Sin embargo, en lugar de simplemente añadir la imagen interpolada a la secuencia de vídeo, un decodificador de vídeo puede usar la trama interpolada durante la decodificación. Por ejemplo, el decodificador de vídeo puede decodificar datos de una imagen actual en base a al menos una parte de la imagen interpolada. En algunos casos, el decodificador de vídeo puede establecer la imagen interpolada igual a la imagen actual. Por ejemplo, el decodificador de vídeo puede decodificar datos de sintaxis para la imagen actual que está incluida en el flujo de bits (por ejemplo, datos de cabecera de fragmento y similares), interpolar la imagen y establecer la imagen interpolada como la imagen actual. En otros casos, el decodificador de vídeo puede interpolar la imagen y decodificar los datos para la imagen actual en relación con la imagen interpolada. En este caso, el decodificador de vídeo puede añadir la imagen interpolada a una memoria de imágenes de referencia con propósitos de predicción.

De ahí que se pueden usar determinadas técnicas descritas en el presente documento que se refieren a FRUC, en algunos ejemplos, para determinar información de movimiento (por ejemplo, en un proceso de derivación de información de movimiento del lado del decodificador). En otros ejemplos, las técnicas descritas en el presente documento que se refieren a FRUC se pueden usar para interpolar datos de vídeo, por ejemplo, para referencia para codificar datos de vídeo, o para salida.

La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y decodificación de vídeo 10 de ejemplo que puede utilizar técnicas para derivar información de movimiento, realizar divisiones de bloques y/o interpolar datos de vídeo. Como se muestra en la FIG. 1, el sistema 10 incluye un dispositivo de origen 12 que proporciona datos de vídeo codificados que se van a decodificar en un momento posterior por un dispositivo de destino 14. En particular, el dispositivo de origen 12 proporciona los datos de vídeo al dispositivo de destino 14 por medio de un medio legible por ordenador 16. El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender cualquiera de una amplia gama de dispositivos, incluyendo ordenadores de mesa, ordenadores plegables (es decir, portátiles), tabletas electrónicas, cajas de conexión, aparatos telefónicos tales como los denominados teléfonos "inteligentes", los denominados paneles "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, un dispositivo de transmisión continua de vídeo o similares. En algunos casos, el dispositivo de origen 12 y el dispositivo de destino 14 pueden estar equipados para la comunicación inalámbrica.

El dispositivo de destino 14 puede recibir los datos de vídeo codificados que se van a descodificar por medio del medio legible por ordenador 16. El medio legible por ordenador 16 puede comprender cualquier tipo de medio o dispositivo que puede transportar los datos de vídeo codificados desde el dispositivo de origen 12 al dispositivo de destino 14. En un ejemplo, el medio legible por ordenador 16 puede comprender un medio de comunicación para habilitar el dispositivo de origen 12 para transmitir datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real. Los datos de vídeo codificados se pueden modular de acuerdo con una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitir al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrica o alámbrica, tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión físicas. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área amplia o una red global, tal como Internet. El medio de comunicación puede incluir encaminadores, conmutadores, estaciones base o cualquier otro equipo que puede ser útil para facilitar la comunicación desde el dispositivo de origen 12 al dispositivo de destino 14.

En algunos ejemplos, se pueden emitir datos codificados desde la interfaz de salida 22 hasta un dispositivo de almacenamiento. De forma similar, se puede acceder a los datos codificados desde el dispositivo de almacenamiento por una interfaz de entrada. El dispositivo de almacenamiento puede incluir cualquiera de una variedad de medios de almacenamiento de datos, distribuidos o de acceso local, tales como un disco duro, discos Blu-ray, DVD, CD-ROM, memoria flash, memoria volátil o no volátil o cualquier otro medio de almacenamiento digital adecuado para almacenar datos de vídeo codificados. En otro ejemplo, el dispositivo de almacenamiento puede corresponder a un servidor de ficheros o a otro dispositivo de almacenamiento intermedio que pueda almacenar el vídeo codificado generado por el dispositivo de origen 12. El dispositivo de destino 14 puede acceder a datos de vídeo almacenados desde el dispositivo de almacenamiento por medio de transmisión continua o descarga. El servidor de ficheros puede ser cualquier tipo de servidor que puede almacenar datos de vídeo codificados y transmitir esos datos de vídeo codificados al dispositivo de destino 14. Los servidores de ficheros de ejemplo incluyen un servidor web (por ejemplo, para un sitio web), un servidor FTP, dispositivos de almacenamiento unido a red (NAS) o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de cualquier conexión de datos estándar, incluyendo una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión wifi), una conexión alámbrica (por ejemplo, DSL, módem de cable, etc.) o una combinación de ambos que sea adecuada para acceder a datos de vídeo codificados almacenados en un servidor de ficheros. La transmisión de datos de vídeo codificados desde el dispositivo de almacenamiento puede ser una transmisión continua, una transmisión de descarga o una combinación de las mismas.

Las técnicas de la presente divulgación no están limitadas necesariamente a aplicaciones o configuraciones inalámbricas. Las técnicas se pueden aplicar a la codificación de vídeo para admitir cualquiera de una variedad de aplicaciones de multimedia, tales como radiodifusiones de televisión por el aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones continuas de vídeo por internet, tales como la transmisión continua adaptativa dinámica sobre HTTP (DASH), vídeo digital que se codifica en un medio de almacenamiento de datos, descodificación de vídeo digital almacenado en un medio de almacenamiento de datos u otras aplicaciones. En algunos ejemplos, se puede configurar el sistema 10 para admitir una transmisión de vídeo unidireccional o bidireccional para admitir aplicaciones tales como transmisión continua de vídeo, reproducción de vídeo, radiodifusión de vídeo y/o videotelefonía.

En el ejemplo de la FIG. 1, el dispositivo de origen 12 incluye una fuente de vídeo 18, un codificador de vídeo 20 y una interfaz de salida 22. El dispositivo de destino 14 incluye una interfaz de entrada 28, un descodificador de vídeo 30 y un dispositivo de visualización 32. De acuerdo con la presente divulgación, el codificador de vídeo 20 del dispositivo fuente 12 se puede configurar para aplicar las técnicas para derivar información de movimiento, realizar divisiones de bloques y/o interpolar datos de vídeo. 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 simplemente un ejemplo. Las técnicas para derivar información de movimiento, realizar divisiones de bloques y/o interpolar datos de vídeo se pueden realizar por cualquier dispositivo de codificación y/o descodificación de vídeo digital. Aunque, en general, las técnicas de la presente divulgación se realizan por un dispositivo de codificación de vídeo, las técnicas también se pueden realizar por un codificador/descodificador de vídeo, denominado típicamente "CÓDEC". Además, las técnicas de la presente divulgación también se pueden realizar mediante un preprocesador de vídeo. El dispositivo de origen 12 y el dispositivo de destino 14 son simplemente ejemplos de dichos dispositivos de codificación, en los que 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. De ahí que el sistema 10 pueda admitir una transmisión de vídeo unidireccional o bidireccional entre los dispositivos de vídeo 12, 14, por ejemplo, para transmisión continua de vídeo, reproducción de vídeo, radiodifusión de vídeo o videotelefonía.

La fuente de vídeo 18 del dispositivo de origen 12 puede incluir un dispositivo de captación de vídeo, tal como una cámara de vídeo, un archivo de vídeo que contiene vídeo captado previamente y/o una interfaz de suministro de vídeo para recibir vídeo desde un proveedor de contenidos 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 la presente 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 vídeo captado, precaptado o generado por ordenador se puede codificar por el codificador de vídeo 20. A continuación, la información de vídeo codificada se puede emitir por la interfaz de salida 22 a un medio legible por ordenador 16.

El medio legible por ordenador 16 puede incluir medios transitorios, tales como una radiodifusión inalámbrica o una transmisión de red alámbrica, o medios de almacenamiento (es decir, medios de almacenamiento no transitorios), tales como un disco duro, una unidad de memoria flash, un disco compacto, un disco de vídeo digital, un disco Blu-ray u otros medios legibles por ordenador. En algunos ejemplos, un servidor de red (no mostrado) puede recibir datos de vídeo codificados desde el dispositivo de origen 12 y proporcionar los datos de vídeo codificados al dispositivo de destino 14, por ejemplo, por medio de transmisión por red. De forma similar, un dispositivo informático de una instalación de producción de medios, tal como una instalación de acuñación de discos, puede recibir datos de vídeo codificados desde el dispositivo de origen 12 y producir un disco que contiene los datos de vídeo codificados. Por lo tanto, se puede entender que el medio legible por ordenador 16 incluye uno o más medios legibles por ordenador de diversas formas, en diversos ejemplos.

La interfaz de entrada 28 del dispositivo de destino 14 recibe información desde el medio legible por ordenador 16. La información del medio legible por ordenador 16 puede incluir información sintáctica definida por el codificador de vídeo 20, que también se usa por el descodificador de vídeo 30, que incluye elementos de sintaxis que describen características y/o procesamiento de bloques y otras unidades codificadas, por ejemplo, los GOP. El dispositivo de visualización 32 presenta 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.

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. Si es aplicable, las unidades MUX-DEMUX se pueden ajustar al protocolo de multiplexor ITU H.223 u otros protocolos tales como el protocolo de datagramas de usuario (UDP).

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. Cuando las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio no transitorio legible por ordenador adecuado y ejecutar las instrucciones en hardware usando uno o más procesadores para realizar las técnicas de la presente divulgación. Tanto el codificador de vídeo 20 como el descodificador de vídeo 30 pueden estar incluidos en uno o más codificadores o descodificadores, de los que cualquiera puede estar integrado como parte de un codificador/descodificador (CÓDEC) combinado en un dispositivo respectivo.

La presente divulgación se puede referir, en general, al codificador de vídeo 20 que "señaliza" determinada información a otro dispositivo, tal como el descodificador de vídeo 30. El término "señalizar" se puede referir, en general, a la comunicación de elementos de sintaxis y/o a otros datos usados para descodificar los datos de vídeo comprimidos. Dicha comunicación se puede producir en tiempo real o casi real. De forma alternativa, dicha comunicación se puede producir durante un período de tiempo, tal como se podría producir cuando se almacenan elementos de sintaxis en un medio de almacenamiento legible por ordenador en un flujo de bits codificado en el momento de la codificación, que, a continuación, se pueden recuperar por un dispositivo de descodificación en cualquier momento después de almacenarse en este medio.

El codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo a una norma de codificación de vídeo. Las normas de codificación de vídeo de ejemplo desarrolladas por el Equipo de colaboración conjunta sobre codificación de vídeo (JCT-VC) y el Equipo de colaboración conjunta sobre el desarrollo de extensiones de codificación de vídeo 3D (JCT-3V) del Grupo de expertos en codificación de vídeo (VCEG) del ITU-T y el Grupo de expertos en imágenes en movimiento (MPEG) de ISO/IEC incluyen la codificación de vídeo de alta eficacia (HEVC) o ITU-T H.265, incluyendo su extensión de alcance, extensión de vista múltiple (MV-HEVC) y extensión

escalable (SHVC). El documento de la norma de HEVC finalizado se publica como "ITU-T H.265, SERIE H: SISTEMAS AUDIOVISUALES Y MULTIMEDIA Infraestructura de servicios audiovisuales - Codificación de vídeo en movimiento - Codificación de vídeo de alta eficacia, "Sector de Normalización de las Telecomunicaciones de la Unión Internacional de Telecomunicaciones (UIT), abril de 2013. De forma alternativa, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con otras normas patentadas o de la industria, tales como ISO/IEC MPEG-4 visual e ITU-T H.264 (también conocida como ISO/IEC MPEG-4 AVC), incluyendo su extensión de codificación de vídeo escalable (SVC) y extensión de codificación de vídeo de vista múltiple (MVC). Sin embargo, las técnicas de la presente divulgación no se limitan a ninguna norma de codificación particular. Por ejemplo, las técnicas de la presente divulgación se pueden usar con una variedad de otras técnicas de codificación de vídeo patentadas o no patentadas o normas posteriores, tales como ITU-T H.266.

Como se indica anteriormente, para los modos de interpredicción, el codificador de vídeo 20 puede buscar un bloque similar al que se está codificando (un "bloque actual") en una imagen de otra ubicación temporal, denominada imagen de referencia. La información usada para identificar la imagen de referencia se puede denominar información de movimiento. Por ejemplo, para cada bloque, puede estar disponible un conjunto de información de movimiento. Un conjunto de información de movimiento contiene información de movimiento para las direcciones de predicción hacia delante y hacia atrás. Aquí, las direcciones de predicción hacia delante y hacia atrás son dos direcciones de predicción de un modo de predicción bidireccional y los términos "hacia delante" y "hacia atrás" no tienen necesariamente un significado geométrico; en cambio, corresponden a la lista de imágenes de referencia 0 (RefPicList0) y a la lista de imágenes de referencia 1 (RefPicList1) de una imagen actual. Cuando solo está disponible una lista de imágenes de referencia para una imagen o un fragmento, solo está disponible la RefPicList0 y la información de movimiento de cada bloque de un fragmento siempre es hacia delante.

En algunos casos, en procesos de descodificación se usa un vector de movimiento conjuntamente con su índice de referencia, un vector de movimiento de este tipo con el índice de referencia asociado se indica como un conjunto unipredictivo de información de movimiento.

Para cada dirección de predicción, la información de movimiento debe contener un índice de referencia y un vector de movimiento. En algunos casos, por simplicidad, se puede hacer referencia a un vector de movimiento por sí mismo, de manera que se suponga que tiene un índice de referencia asociado. Se usa un índice de referencia para identificar una imagen de referencia en la lista de imágenes de referencia actuales (RefPicList0 o RefPicList1). Un vector de movimiento tiene una componente horizontal y otro vertical.

El recuento de orden de imágenes (POC) se usa ampliamente en las normas de codificación de vídeo para identificar un orden de visualización de una imagen. Aunque existen casos en que dos imágenes dentro de una secuencia de vídeo codificada pueden tener el mismo valor de POC, típicamente no ocurre dentro de una secuencia de vídeo codificada. Cuando están presentes múltiples secuencias de vídeo codificado en un flujo de bits, las imágenes con un mismo valor de POC pueden estar más cercanas entre sí en términos de orden de descodificación. Los valores de POC de las imágenes se usan típicamente para la construcción de la lista de imágenes de referencia, la derivación de un conjunto de imágenes de referencia como en la HEVC y el escalado del vector de movimiento.

En H.264/AVC, cada macrobloque (MB) inter se puede dividir en cuatro maneras diferentes, que incluyen: una división de MB de 16x16; dos divisiones de MB de 16x8; dos divisiones de MB de 8x16; y cuatro divisiones de MB de 8x8. Diferentes divisiones de MB en un MB pueden tener diferentes valores de índice de referencia para cada dirección (RefPicList0 o RefPicList1). Cuando un MB no está dividido en cuatro divisiones de MB de 8x8, tiene solo un vector de movimiento para cada división de MB en cada dirección.

Cuando un MB se divide en cuatro divisiones de MB de 8x8, cada división de 8x8 MB se puede dividir además en subbloques, de los que cada uno puede tener un vector de movimiento diferente en cada dirección. Existen cuatro maneras diferentes de obtener subbloques a partir de una división de MB de 8x8: un subbloque de 8x8, dos subbloques de 8x4, dos subbloques de 4x8; y cuatro subbloques de 4x4. Cada subbloque puede tener un vector de movimiento diferente en cada dirección. Por lo tanto, el vector de movimiento está presente en un nivel de igual a mayor que el subbloque.

En AVC, el modo directo temporal se puede habilitar en el nivel de MB o bien de división de MB para el modo de salto o directo en los fragmentos B. Para cada división de MB, los vectores de movimiento del bloque coubicado con la división de MB actual en la RefPicList1 [ 0 ] del bloque actual se usan para derivar los vectores de movimiento. Cada vector de movimiento en el bloque coubicado se ajusta a escala en base a las distancias de POC. En AVC, un modo directo también puede predecir la información de movimiento desde los vecinos espaciales, lo que se puede denominar modo directo espacial.

En HEVC, para generar una representación codificada de una imagen, el codificador de vídeo 20 puede generar un conjunto de unidades de árbol de codificación (CTU). Cada una de las CTU puede comprender un bloque de árbol de codificación (CTB) de muestras de luma, dos CTB correspondientes de muestras de croma y estructuras de sintaxis usadas para codificar las muestras de los CTB. En imágenes monocromáticas o imágenes que tienen



tres planos de color separados, una CTU puede comprender un único bloque de CTB y estructuras de sintaxis usadas para codificar las muestras del bloque de árbol de codificación.

Un bloque de árbol de codificación puede ser un bloque de muestras de  $N \times N$ . El tamaño de un CTB puede variar de  $16 \times 16$  a  $64 \times 64$  en el perfil principal de HEVC (aunque, técnicamente, se pueden admitir tamaños de CTB de  $8 \times 8$ ). Aunque una unidad de codificación (CU) podría ser del mismo tamaño que un CTB y tan pequeña como  $8 \times 8$ . Cada unidad de codificación está codificada con un modo. Una CTU también se puede denominar "bloque de árbol" o "unidad de codificación de máximo tamaño" (LCU). Las CTU de HEVC pueden ser análogas en términos generales a los macrobloques de otras normas, tales como H.264/AVC. Sin embargo, una CTU no está necesariamente limitada a un tamaño particular y puede incluir una o más unidades de codificación (CU). Un fragmento puede incluir un número entero de CTU ordenadas consecutivamente en un orden de exploración de barrido.

Para generar una CTU codificada, el codificador de vídeo 20 puede realizar de forma recursiva una división de árbol cuaternario en los bloques de árbol de codificación de una CTU para dividir los bloques de árbol de codificación en bloques de codificación, de ahí el nombre "unidades de árbol de codificación". Un bloque de codificación puede ser un bloque de muestras de  $N \times N$ . Una CU puede comprender un bloque de codificación de muestras de luma y dos bloques de codificación correspondientes de muestras de croma de una imagen que tiene una matriz de muestras de luma, una matriz de muestras Cb y una matriz de muestras Cr y estructuras de sintaxis usadas para codificar las muestras de los bloques de codificación. En imágenes monocromáticas o imágenes que tienen tres planos de color separados, una CU puede comprender un único bloque de codificación y estructuras de sintaxis usadas para codificar las muestras del bloque de codificación.

El codificador de vídeo 20 puede dividir un bloque de codificación de una CU en uno o más bloques de predicción. Un bloque de predicción es un bloque rectangular (es decir, cuadrado o no cuadrado) de muestras en las que se aplica la misma predicción. Una unidad de predicción (PU) de una CU puede comprender un bloque de predicción de muestras de luma, dos bloques de predicción correspondientes de muestras de croma y estructuras de sintaxis usadas para predecir los bloques de predicción. En imágenes monocromáticas o imágenes que tienen tres planos de color separados, una PU puede comprender un único bloque de predicción y estructuras de sintaxis usadas para predecir el bloque de predicción. El codificador de vídeo 20 puede generar bloques predictivos de luma, Cb y Cr para bloques de predicción de luma, Cb y Cr de cada PU de la CU.

El codificador de vídeo 20 puede usar intrapredicción o interpredicción para generar los bloques predictivos para una PU. Si el codificador de vídeo 20 usa intrapredicción para generar los bloques predictivos de una PU, el codificador de vídeo 20 puede generar los bloques predictivos de la PU en base a muestras descodificadas de la imagen asociada con la PU. Si el codificador de vídeo 20 usa interpredicción para generar los bloques predictivos de una PU, el codificador de vídeo 20 puede generar los bloques predictivos de la PU en base a muestras descodificadas de una o más imágenes distintas a la imagen asociada con la PU. Cuando se intercodifica una CU, la CU se puede dividir además en dos o cuatro PU. Cuando dos PU están presentes en una CU, las PU pueden ser en algunos casos rectángulos de la mitad del tamaño o del tamaño de dos rectángulos con un cuarto o tres cuartos del tamaño de la CU.

Después de que el codificador de vídeo 20 genera bloques predictivos de luma, Cb y Cr para una o más PU de una CU, el codificador de vídeo 20 puede generar un bloque residual de luma para la CU. Cada muestra en el bloque residual de luma de la CU indica una diferencia entre una muestra de luma en uno de los bloques predictivos de luma de la CU y una muestra correspondiente en el bloque de codificación de luma original de la CU. Además, el codificador de vídeo 20 puede generar un bloque residual de Cb para la CU. Cada muestra en el bloque residual de Cb de la CU puede indicar una diferencia entre una muestra de Cb en uno de los bloques predictivos de Cb de la CU y una muestra correspondiente en el bloque de codificación de Cb original de la CU. El codificador de vídeo 20 también puede generar un bloque residual de Cr para la CU. Cada muestra en el bloque residual de Cr de la CU puede indicar una diferencia entre una muestra de Cr en uno de los bloques predictivos de Cr de la CU y una muestra correspondiente en el bloque de codificación de Cr original de la CU.

Además, el codificador de vídeo 20 puede usar una división de árbol cuaternario para descomponer los bloques residuales de luma, Cb y Cr de una CU en uno o más bloques de transformada de luma, Cb y Cr. Un bloque de transformada es un bloque rectangular (por ejemplo, cuadrado o no cuadrado) de muestras a las que se aplica la misma transformada. Una unidad de transformada (TU) de una CU puede comprender un bloque de transformada de muestras de luma, dos bloques correspondientes de transformada de muestras de croma y estructuras de sintaxis usadas para transformar las muestras de bloques de transformada. De este modo, cada TU de una CU puede estar asociada con un bloque de transformada de luma, un bloque de transformada de Cb y un bloque de transformada de Cr. El bloque de transformada de luma asociado con la TU puede ser un subbloque del bloque residual de luma de la CU. El bloque de transformada de Cb puede ser un subbloque del bloque residual de Cb de la CU. El bloque de transformada de Cr puede ser un subbloque del bloque residual de Cr de la CU. En imágenes monocromáticas o imágenes que tienen tres planos de color separados, una TU puede comprender un único bloque de transformada y estructuras de sintaxis usadas para transformar las muestras del bloque de transformada.

El codificador de vídeo 20 puede aplicar una o más transformadas a un bloque de transformada de luma de una TU para generar un bloque de coeficientes de luma para la TU. Un bloque de coeficientes puede ser una matriz bidimensional de coeficientes de transformada. Un coeficiente de transformada puede ser una cantidad escalar. El codificador de vídeo 20 puede aplicar una o más transformadas a un bloque de transformada de Cb de una TU para generar un bloque de coeficientes de Cb para la TU. El codificador de vídeo 20 puede aplicar una o más transformadas a un bloque de transformada de Cr de una TU para generar un bloque de coeficientes de Cr para la TU.

Después de generar un bloque de coeficientes (por ejemplo, un bloque de coeficientes de luma, un bloque de coeficientes de Cb o un bloque de coeficientes de Cr), el codificador de vídeo 20 puede cuantificar el bloque de coeficientes. La cuantificación se refiere, en general, a un proceso en el que los coeficientes de transformada se cuantifican para reducir, posiblemente, la cantidad de datos usados para representar los coeficientes de transformada, proporcionando compresión adicional. Después de que el codificador de vídeo 20 cuantifica un bloque de coeficientes, el codificador de vídeo 20 puede codificar por entropía elementos de sintaxis que indican los coeficientes de transformada cuantificados. Por ejemplo, el codificador de vídeo 20 puede realizar una codificación aritmética binaria adaptativa al contexto (CABAC) en los elementos de sintaxis que indican los coeficientes de transformada cuantificados.

El codificador de vídeo 20 puede emitir un flujo de bits que incluye una secuencia de bits que forma una representación de imágenes codificadas y datos asociados. El flujo de bits puede comprender una secuencia de unidades de capa de abstracción de red (NAL). Una unidad de NAL es una estructura de sintaxis que contiene una indicación del tipo de datos en la unidad de NAL y octetos que contienen esos datos en forma de una carga útil de secuencia de octetos sin procesar (RBSP) entremezclados como sea necesario con bits de prevención de emulación. Cada una de las unidades de NAL incluye una cabecera de unidad de NAL y encapsula una RBSP.

Diferentes tipos de unidades de NAL pueden encapsular diferentes tipos de RBSP. Por ejemplo, un primer tipo de unidad de NAL puede encapsular una RBSP para un conjunto de parámetros de imagen (PPS), un segundo tipo de unidad de NAL puede encapsular una RBSP para un fragmento codificado, un tercer tipo de unidad de NAL puede encapsular una RBSP para SEI y así sucesivamente. Las unidades de NAL que encapsulan las RBSP para datos de codificación de vídeo (a diferencia de las RBSP para conjuntos de parámetros y mensajes SEI) se pueden denominar unidades de NAL de la capa de codificación de vídeo (VCL).

El descodificador de vídeo 30 puede recibir un flujo de bits generado por el codificador de vídeo 20. Además, el descodificador de vídeo 30 puede analizar el flujo de bits para obtener elementos de sintaxis a partir del flujo de bits. El descodificador de vídeo 30 puede reconstruir las imágenes de los datos de vídeo en base, al menos en parte, a los elementos de sintaxis obtenidos a partir del flujo de bits. El proceso para reconstruir los datos de vídeo puede ser, en general, recíproco al proceso realizado por el codificador de vídeo 20. Además, el descodificador de vídeo 30 puede realizar una cuantificación inversa de los bloques de coeficientes asociados con las TU de una CU actual. El descodificador de vídeo 30 puede realizar transformadas inversas en los bloques de coeficientes para reconstruir los bloques de transformada asociados con las TU de la CU actual. El descodificador de vídeo 30 puede reconstruir los bloques de codificación de la CU actual añadiendo las muestras de los bloques predictivos para las PU de la CU actual a las muestras correspondientes de los bloques de transformada de las TU de la CU actual. Al reconstruir los bloques de codificación para cada CU de una imagen, el descodificador de vídeo 30 puede reconstruir la imagen.

Cuando una CU se intercodifica, está presente un conjunto de información de movimiento para cada PU. Además, cada PU se codifica con un modo único de interpredicción para derivar un conjunto de información de movimiento. En la norma HEVC, existen dos modos de interpredicción, denominados modos de fusión (el salto se considera un caso especial de fusión) y de predicción de vector de movimiento avanzada (AMVP) respectivamente para una unidad de predicción (PU).

Tanto en el modo AMVP como de fusión, se mantiene una lista de candidatos de vector de movimiento (MV) para predictores de vector de movimiento múltiples. El/los vector(es) de movimiento, así como los índices de referencia en el modo de fusión, de la PU actual se generan tomando un candidato de la lista de candidatos de MV.

La lista de candidatos de MV contiene hasta cinco candidatos para el modo de fusión y dos candidatos para el modo AMVP. Un candidato de fusión puede contener un conjunto de información de movimiento, por ejemplo, vectores de movimiento correspondientes a ambas listas de imágenes de referencia (lista 0 y lista 1) y los índices de referencia. Si un candidato de fusión se identifica por un índice de fusión, las imágenes de referencia se usan para la predicción de los bloques actuales, así como se determinan los vectores de movimiento asociados. Sin embargo, en el modo AMVP, para cada dirección de predicción potencial de la lista 0 o bien la lista 1, se necesita señalar explícitamente un índice de referencia, conjuntamente con un índice de MVP, a la lista de candidatos de MV, puesto que el candidato de AMVP solo contiene un vector de movimiento. En el modo AMVP, los vectores de movimiento predichos se pueden refinar además.

Como se puede observar anteriormente, un candidato de fusión corresponde a un conjunto completo de información de movimiento, mientras que un candidato de AMVP contiene solo un vector de movimiento para una dirección de predicción e índice de referencia específicos. Los candidatos para ambos modos se derivan de forma similar a partir de los mismos bloques vecinos espaciales y temporales, como se describe con respecto a las FIGS. 4 y 5 a continuación.

De acuerdo con los aspectos de la presente divulgación, como se describe con más detalle a continuación, el codificador de vídeo 20 y/o el decodificador de vídeo 30 se pueden configurar para realizar cualquier combinación de las técnicas descritas en el presente documento para derivar información de movimiento, realizar divisiones de bloques y/o interpolar datos de vídeo. Con respecto a la derivación de información de movimiento, el codificador de vídeo 20 y/o el decodificador de vídeo 30 se pueden configurar para derivar información de movimiento realizando una búsqueda de movimiento para un primer conjunto de datos de referencia que corresponde a un segundo conjunto de datos de referencia fuera del bloque actual. La correspondencia se puede determinar en base a una cantidad de similitud entre los datos de referencia, y se puede denominar en el presente documento como determinación de una "coincidencia" o de "la mejor coincidencia".

En algunos ejemplos, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden codificar inicialmente uno o más elementos de sintaxis que indican si está habilitado el proceso de derivación de movimiento. En algunos casos, el uno o más elementos de sintaxis se pueden incorporarse con otro modo, tal como el modo de fusión descrito anteriormente. Por ejemplo, como se describe con más detalle con respecto al ejemplo de la FIG. 10, el codificador de vídeo 20 y/o el decodificador de vídeo 30 codifican uno o más elementos de sintaxis cuando se realiza el modo de fusión (por ejemplo, un indicador, un índice en una lista de candidatos de fusión o similares) que indica si se realiza la derivación de movimiento.

En los casos en los que se habilita la derivación de movimiento, de acuerdo con aspectos de la presente divulgación, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden realizar la derivación de vector de movimiento con patrón coincidente. Por ejemplo, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden codificar uno o más elementos de sintaxis que indican qué modo de derivación de información de movimiento aplicar a partir de una pluralidad de modos de derivación de información de movimiento. El codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden codificar un indicador para distinguir entre dos modos de información de derivación de movimiento o un índice para distinguir entre más de dos modos de derivación de información de movimiento. Como se describe en el presente documento, los modos de derivación de información de movimiento con patrón coincidente de ejemplo incluyen coincidencia bilateral o coincidencia de plantillas.

Durante el proceso de derivación de movimiento, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden derivar un vector de movimiento inicial para todo un bloque (por ejemplo, una PU completa) en base al proceso de derivación de movimiento seleccionado. Por ejemplo, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden usar vectores de movimiento de una lista de candidatos asociada con el modo de fusión y determinar qué vector de movimiento de la lista de candidatos da como resultado la mejor coincidencia. Es decir, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden determinar qué vector de movimiento de la lista de candidatos, cuando se usa en el proceso de derivación de movimiento seleccionado, da como resultado un primer conjunto de datos de referencia que corresponde a un segundo conjunto de datos de referencia fuera del bloque actual, por ejemplo, datos de referencia que coinciden de forma estrecha con los datos en la imagen actual u otra imagen de referencia. En general, "la mejor coincidencia" se puede referir a datos de vídeo que son los más similares en términos de diferencias de píxeles.

Como ejemplo con propósitos ilustrativos, como se describe con más detalle con respecto a la FIG. 8, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden seleccionar un modo de derivación de información de movimiento por coincidencia de plantillas. En este ejemplo, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden seleccionar el candidato de vector de movimiento a partir del modo de fusión en base al mejor candidato de vector que da como resultado una plantilla que coincide de forma lo más estrecha con una plantilla en una imagen de referencia. Por ejemplo, el codificador de vídeo 20 y/o el decodificador de vídeo 30 se pueden configurar para realizar una búsqueda de movimiento para un primer conjunto de datos de referencia que corresponde a un segundo conjunto de datos de referencia fuera del bloque actual, donde el primer conjunto de datos comprende la plantilla en la imagen actual y el segundo conjunto de datos de referencia comprende la plantilla en la imagen de referencia. En algunos casos, como se describe con más detalle a continuación, el codificador de vídeo 20 y/o el decodificador de vídeo 30 puede seleccionar de forma adicional o alternativa un vector de movimiento candidato en base a un coste de coincidencia mínimo.

De acuerdo con aspectos de la presente divulgación, después de determinar el vector de movimiento candidato, el codificador de vídeo 20 y/o el decodificador de vídeo 30 puede refinar además el vector de movimiento candidato para derivar la información de movimiento para el bloque actual. Por ejemplo, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden realizar una búsqueda (por ejemplo, una búsqueda usando el modo de derivación de información de movimiento) en un área predefinida de la imagen de referencia indicada por el vector de movimiento candidato. El codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden derivar la información de movimiento para el bloque actual en base al vector de movimiento que indica datos de referencia en el área

predefinida que tienen una mejor coincidencia y/o un coste de coincidencia mínimo con los datos de vídeo de la imagen actual.

En algunos casos, de acuerdo con aspectos de la presente divulgación, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden dividir el bloque de datos de vídeo en subbloques cuando se deriva información de movimiento. Por ejemplo, el codificador de vídeo 20 y/o el descodificador de vídeo 30 puede derivar por separado información de movimiento para cada subbloque de un bloque más grande. En algunos casos, el codificador de vídeo 20 y/o el descodificador de vídeo 30 puede usar la información de movimiento derivada para un bloque (por ejemplo, usando el proceso descrito anteriormente) como información de movimiento candidata para cada uno de los subbloques. A continuación, el codificador de vídeo 20 y/o el descodificador de vídeo 30 puede refinar además la información de movimiento derivada para cada uno de los subbloques, por ejemplo, usando un modo de derivación de información de movimiento particular (por ejemplo, una técnica de coincidencia bilateral, una técnica de coincidencia de plantillas u otra técnica, como se describe con más detalle a continuación).

La FIG. 2 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo 20 que puede implementar técnicas para derivar información de movimiento, realizar divisiones de bloques y/o interpolar datos de vídeo. El codificador de vídeo 20 puede realizar intracodificación e intercodificación de bloques de vídeo dentro de fragmentos de vídeo. La intracodificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial de un vídeo dentro de una trama o imagen de vídeo dada. La intercodificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal de un vídeo dentro de tramas o imágenes contiguas de una secuencia de vídeo. El intramodo (modo I) se puede referir a cualquiera de varios modos de codificación de base espacial. Los intermodos, tales como la predicción unidireccional (modo P) o la predicción bidireccional (modo B), se pueden referir a cualquiera de varios modos de codificación de base temporal.

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 selección de modo 40, una memoria de imágenes de referencia 64, un sumador 50, una unidad de procesamiento de transformada 52, una unidad de cuantificación 54 y una unidad de codificación entrópica 56. A su vez, la unidad de selección de modo 40 incluye una unidad de compensación de movimiento 44, una unidad de estimación de movimiento 42, una unidad de intrapredicción 46 y una unidad de división 48. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 también incluye 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 mostrado en la FIG. 2) para filtrar 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. También se pueden usar filtros adicionales (en bucle o posteriormente al bucle), además del filtro de eliminación de bloques. Dichos filtros no se muestran por razones de brevedad, pero si se desea, pueden filtrar la salida del sumador 50 (como un filtro en bucle).

Durante el proceso de codificación, el codificador de vídeo 20 recibe una trama o fragmento de vídeo que se va a codificar. La trama o fragmento se pueden dividir en múltiples bloques de vídeo. La memoria de datos de vídeo 38 puede almacenar los datos de vídeo que se van a codificar por 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. La memoria de imágenes de referencia 64 se puede denominar un DPB que almacena datos de vídeo de referencia para su uso en la codificación de datos de vídeo por el codificador de vídeo 20, por ejemplo, en modos de intracodificación o intercodificación. La memoria de datos de vídeo 38 y la memoria de imágenes de referencia 64 se pueden formar por cualquiera de una variedad de dispositivos de memoria, tales como memoria dinámica de acceso aleatorio (DRAM), incluyendo DRAM síncrona (SDRAM), RAM magnetorresistiva (MRAM), RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. La memoria de datos de vídeo 38 y la memoria de imágenes de referencia 64 se pueden proporcionar por el mismo dispositivo de memoria o por dispositivos de memoria separados. 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.

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

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

La unidad de selección de modo 40 puede seleccionar uno de los modos de codificación, intra o inter, por ejemplo, en base a 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. La unidad de selección de modo 40 también proporciona elementos de sintaxis, tales como vectores de movimiento, indicadores de intramodo, información de división y otra información de sintaxis de este tipo a la unidad de codificación entrópica 56.

La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden estar altamente integradas, pero se ilustran por separado con propósitos conceptuales. La estimación de movimiento, realizada por la unidad de estimación de movimiento 42, es el proceso de generación de vectores de movimiento, que estiman el movimiento para los bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de una PU de un bloque de vídeo dentro de una trama o imagen de vídeo actual, en relación con un bloque predictivo dentro de una trama de referencia (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 encuentra que coincide de forma estrecha con el bloque que se va a codificar, en términos de la diferencia de píxeles, lo que se puede determinar por una suma de diferencias absolutas (SAD), una suma de diferencias al cuadrado (SSD) u otras métricas de diferencia.

En algunos ejemplos, el codificador de vídeo 20 puede realizar una búsqueda de movimiento rápido para determinar un vector de movimiento de un bloque. Existen muchos procedimientos de búsqueda de movimiento rápido propuestos en la literatura, tales como la búsqueda en descenso de gradiente basada en bloques (BBGDS) como se describe, por ejemplo, en Lurng-Kuo Liu, Ephraim Feig, "A block-based gradient descent search algorithm for block motion estimation in video coding", IEEE Trans. Circuits Syst. Video Technol., vol. 6, págs., 419-422, agosto de 1996, búsqueda de rombos con sesgo central sin restricciones (UCBDS), como se describe, por ejemplo, en Jo Yew Tham, Surendra Ranganath, Maitreya Ranganath y Ashraf Ali Kassim, "A novel unrestricted center-biased diamond search algorithm for block motion estimation", IEEE Trans. Circuits Syst. Video Technol., vol. 8, págs. 369-377, agosto de 1998, y búsqueda basada en hexágonos (HEBS) como se describe, por ejemplo, en Ce Zhu, Xiao Lin y Lap-Pui Chau, "Hexagon-Based Search Pattern for Fast Block Motion Estimation", IEEE Trans. Circuits Syst. Video Technol., vol. 12, págs. 349-355, mayo de 2002. Básicamente, estas técnicas incluyen buscar solo un determinado número de posiciones en el interior de una ventana de búsqueda en base a patrones de búsqueda predefinidos. Estas técnicas habitualmente funcionan bien cuando el movimiento es pequeño y moderado.

En algunos ejemplos, el codificador de vídeo 20 puede calcular valores para posiciones de píxel fraccionario de imágenes de referencia almacenadas en la memoria de imágenes de referencia 64. Por ejemplo, el codificador de vídeo 20 puede interpolar valores de posiciones de un cuarto de píxel, posiciones de un octavo de píxel u otras posiciones de píxel fraccionario de la imagen de referencia. Por lo tanto, la unidad de estimación de movimiento 42 puede realizar una búsqueda de movimiento en relación con las posiciones de píxel completo y las posiciones de píxel fraccionario y emitir un vector de movimiento con una precisión de píxel fraccionario.

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

La compensación de movimiento, realizada por la unidad de compensación de movimiento 44, puede implicar buscar o generar el bloque predictivo en base al vector de movimiento determinado por la unidad de estimación de movimiento 42. De nuevo, la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 se pueden integrar funcionalmente, en algunos ejemplos. Tras recibir el vector de movimiento para la PU del bloque de vídeo actual, la unidad de compensación de movimiento 44 puede ubicar el bloque predictivo al que apunta el vector de movimiento en una de las listas de imágenes de referencia. El sumador 50 forma un bloque de vídeo residual restando valores de píxel del bloque predictivo a los valores de píxel del bloque de vídeo actual que se está codificando, formando valores de diferencias de píxel, como se analiza a continuación. En general, la unidad de estimación de movimiento 42 realiza la estimación de movimiento en relación con los componentes de luma, y la unidad de compensación de movimiento 44 usa los vectores de movimiento calculados en base a los componentes de luma tanto para los componentes de croma como para los componentes de luma. La unidad de selección de modo 40 también puede generar elementos de sintaxis asociados con los bloques de vídeo y el fragmento de vídeo para su uso por el descodificador de vídeo 30 en la descodificación de los bloques de vídeo del fragmento de vídeo.

La unidad de intrapredicción 46 puede intraprededir un bloque actual, como una alternativa a la interpredicción realizada por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44, como se

describe anteriormente. En particular, la unidad de intrapredicción 46 puede determinar un modo de intrapredicción para usar para codificar un bloque actual. En algunos ejemplos, la unidad de intrapredicción 46 puede codificar un bloque actual usando diversos modos de intrapredicción, por ejemplo, durante pases de codificación separados, y la unidad de intrapredicción 46 (o unidad de selección de modo 40, en algunos ejemplos) puede seleccionar un modo de intrapredicción apropiado para usar a partir de los modos sometidos a prueba.

Por ejemplo, la unidad de intrapredicción 46 puede calcular valores de velocidad-distorsión usando un análisis de velocidad-distorsión para los diversos modos de intrapredicción sometidos a prueba, y seleccionar el modo de intrapredicción que tiene las mejores características de velocidad-distorsión entre los modos sometidos a prueba. El análisis de velocidad-distorsión determina, en general, una cantidad de distorsión (o de error) entre un bloque codificado y un bloque original, no codificado, que se codificó para producir el bloque codificado, así como una velocidad de transferencia de bits (es decir, un número de bits) usada para producir el bloque codificado. La unidad de intrapredicción 46 puede calcular proporciones a partir de las distorsiones y velocidades para los diversos bloques codificados para determinar qué modo de intrapredicción presenta el mejor valor de velocidad-distorsión para el bloque.

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

De acuerdo con aspectos de la presente divulgación, como se describe en el presente documento, el codificador de vídeo 20 se puede configurar para realizar cualquier combinación de las técnicas descritas en el presente documento para derivar información de movimiento, realizar divisiones de bloques y/o interpolar datos de vídeo. En particular, determinadas técnicas de la presente divulgación se pueden realizar por la unidad de derivación 49. Por ejemplo, la unidad de derivación 49 se puede configurar para determinar información de movimiento para un bloque actual y sin incluir datos que indican la información de movimiento en el flujo de bits.

En algunos casos, la unidad de derivación 49 (y/o la unidad de selección de modo 40) puede determinar si realizar la derivación de movimiento para un bloque particular (por ejemplo, frente a la intrapredicción o la interpredicción tradicional) en base a un análisis de distorsión de velocidad. Por ejemplo, la unidad de derivación 49 puede determinar si se realiza la derivación de movimiento de una manera similar a una selección de coste de distorsión de velocidad como se realiza para los candidatos de fusión en el modo de fusión. En este ejemplo, la unidad de derivación 49 puede verificar cada modo de derivación de información de movimiento de una pluralidad de modos de derivación de información de movimiento (por ejemplo, un modo de coincidencia bilateral, modo de coincidencia de plantillas o similares) usando una selección de coste de distorsión de velocidad. La unidad de derivación 49 puede comparar además el modo de derivación de información de movimiento que tiene el coste mínimo con otros modos de PU (por ejemplo, modos de intrapredicción, un modo de interpredicción tradicional, un modo de codificación de paleta o similares). En los casos en los que el modo de derivación de movimiento es el modo más eficaz en términos de eficacia de codificación, el codificador de vídeo 20 puede codificar uno o más elementos de sintaxis que indican que se deriva (en lugar de señalar) la información de movimiento para el bloque actual. El codificador de vídeo 20 también puede codificar uno o más elementos de sintaxis para la indicación del modo de derivación de movimiento a partir de una pluralidad de modos de derivación de información de movimiento.

En otros ejemplos, de acuerdo con aspectos de la presente divulgación, la unidad de derivación 49 puede interpolar datos de vídeo que no están incluidos en el flujo de bits codificado para una secuencia de vídeo. Por ejemplo, la unidad de derivación 49 puede realizar cualquier combinación de técnicas de derivación de movimiento para interpolar una imagen que no está incluida en el flujo de bits, por ejemplo, similar a la conversión ascendente de velocidad de tramas. En algunos casos, el codificador de vídeo 20 puede usar la imagen interpolada durante la codificación. Por ejemplo, la unidad de derivación 49 puede interpolar una imagen y el codificador de vídeo 20 puede codificar los datos para una imagen actual en relación con la imagen interpolada. En este ejemplo, el codificador de vídeo 20 puede añadir la imagen interpolada a la memoria de imágenes de referencia 64 y codificar los datos de otras imágenes en base a al menos una parte de la imagen interpolada.

En otros ejemplos, la unidad de derivación 49 puede interpolar una imagen y el codificador de vídeo 20 puede establecer la imagen interpolada igual a la imagen actual. Por ejemplo, la unidad de derivación 49 puede interpolar la imagen actual y el codificador de vídeo 20 puede codificar los datos de sintaxis para que la imagen actual se incluya en el flujo de bits (por ejemplo, datos de cabecera de fragmento y similares), pero puede omitir la codificación de datos de vídeo para la imagen actual.

Si bien la unidad de derivación 49 se puede configurar para realizar determinadas técnicas de derivación y/o interpolación, como se describe en el presente documento, se debe entender que una o más de otras unidades del codificador de vídeo 20 también o de forma alternativa se pueden configurar para interpolar los datos. Por ejemplo, el codificador de vídeo 20 puede incluir una variedad de otros interpoladores o filtros, por ejemplo, para interpolar un píxel en una ubicación de subpíxel (subpíxel) durante una compensación de movimiento.

El codificador de vídeo 20 forma un bloque de vídeo residual restando los datos de predicción de la unidad de selección de modo 40 al bloque de vídeo original que se está codificando. El sumador 50 representa el componente o los componentes que realizan esta operación de resta. La unidad de procesamiento de transformada 52 aplica una transformada, tal como una transformada de coseno discreta (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 procesamiento de transformada 52 puede realizar otras transformadas que son conceptualmente similares a la DCT. También se podrían usar transformadas de ondículas, transformadas de enteros, transformadas de subbanda u otros tipos de transformadas.

En cualquier caso, la unidad de procesamiento 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 frecuencia. La unidad de procesamiento de transformada 52 puede enviar los coeficientes de transformada resultantes a la unidad de cuantificación 54. La unidad de cuantificación 54 cuantifica los coeficientes de transformada para reducir además la velocidad de transferencia de bits. El proceso de cuantificación puede reducir la profundidad de bits asociada con algunos o todos los coeficientes. El grado de cuantificación se puede modificar ajustando un parámetro de cuantificación. En algunos ejemplos, la unidad de cuantificación 54 puede realizar, a continuación, una exploración de la matriz que incluye los coeficientes de transformada cuantificados. De forma alternativa, la unidad de codificación entrópica 56 puede realizar la exploración.

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

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 de la memoria de imágenes de referencia 64. La unidad de compensación de movimiento 44 también puede aplicar uno o más filtros de interpolación al bloque residual reconstruido para calcular valores de píxeles fraccionarios, 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 la memoria de imágenes de referencia 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 ejemplo de descodificador de vídeo 30 que puede implementar técnicas para derivar información de movimiento, realizar divisiones de bloques y/o interpolar datos de vídeo. En el ejemplo de la FIG. 3, el descodificador de vídeo 30 incluye la memoria de datos de vídeo 68, una unidad de descodificación entrópica 70, la unidad de compensación de movimiento 72, la unidad de intrapredicción 74, la unidad de cuantificación inversa 76, la unidad de transformación inversa 78, la memoria de imágenes de referencia 82 y el sumador 80. En algunos ejemplos, el descodificador de vídeo 30 puede realizar un pase de descodificación, en general, recíproco al pase de codificación descrito con respecto al codificador de vídeo 20 (FIG. 2). La unidad de compensación de movimiento 72 puede generar datos de predicción en base a vectores de movimiento recibidos desde la unidad de descodificación entrópica 70, mientras que la unidad de intrapredicción 74 puede generar datos de predicción en base a indicadores de modo de intrapredicción recibidos desde la unidad de descodificación entrópica 70.

Durante el proceso de descodificación, el descodificador de vídeo 30 recibe un flujo de bits de vídeo codificado que representa bloques de vídeo de un fragmento de vídeo codificado y elementos de sintaxis asociados desde el codificador de vídeo 20. Los datos de vídeo almacenados en la memoria de datos de vídeo 68 se pueden obtener, por ejemplo, de un medio legible por ordenador, por ejemplo, de una fuente de vídeo local, tal como una cámara, por medio de comunicación de datos de vídeo por red alámbrica o inalámbrica, o accediendo a medios físicos de almacenamiento de datos. La memoria de datos de vídeo 68 puede formar una memoria intermedia de imágenes codificadas (CPB) que almacena datos de vídeo codificados a partir de un flujo de bits de vídeo codificado.

La memoria de imágenes de referencia 82 se puede denominar DPB que almacena datos de vídeo de referencia para su uso en la decodificación de datos de vídeo por el decodificador de vídeo 30, por ejemplo, en modos de intracodificación o intercodificación. La memoria de datos de vídeo 68 y la memoria de imágenes de referencia 82 se pueden formar por cualquiera de una variedad de dispositivos de memoria, tales como memoria dinámica de acceso aleatorio (DRAM), incluyendo DRAM síncrona (SDRAM), RAM magnetorresistiva (MRAM), RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. La memoria de datos de vídeo 68 y la memoria de imágenes de referencia 82 se pueden proporcionar por el mismo dispositivo de memoria o por dispositivos de memoria separados. En diversos ejemplos, la memoria de datos de vídeo 68 puede estar en un chip con otros componentes del decodificador de vídeo 30, o fuera del chip en relación con esos componentes.

La unidad de decodificación entrópica 70 del decodificador de vídeo 30 decodifica por entropía el flujo de bits para generar coeficientes cuantificados, vectores de movimiento o indicadores de modo de intrapredicción y otros elementos de sintaxis. La unidad de decodificación entrópica 70 reenvía los vectores de movimiento y otros elementos de sintaxis a la unidad de compensación de movimiento 72. El decodificador de vídeo 30 puede recibir los elementos de sintaxis a nivel de fragmento de vídeo y/o a nivel de bloque de vídeo.

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

La unidad de compensación de movimiento 72 determina información de predicción para un bloque de vídeo del fragmento de vídeo actual analizando los vectores de movimiento y otros elementos de sintaxis, y usa la información de predicción para producir los bloques predictivos para el bloque de vídeo actual que se esté decodificando. Por ejemplo, la unidad de compensación de movimiento 72 usa algunos de los elementos de sintaxis recibidos para determinar un modo de predicción (por ejemplo, de intrapredicción o interpredicción) usado para codificar los bloques de vídeo del fragmento de vídeo, un tipo de fragmento de interpredicción (por ejemplo, un fragmento B, un fragmento P o un fragmento GPB), información de construcción para una o más de las listas de imágenes de referencia para el fragmento, vectores de movimiento para cada bloque de vídeo intercodificado del fragmento, un estado de interpredicción para cada bloque de vídeo intercodificado del fragmento y otra información para decodificar los bloques de vídeo del fragmento de vídeo actual.

La unidad de compensación de movimiento 72 también puede realizar interpolación en base a filtros de interpolación. La unidad de compensación de movimiento 72 puede usar filtros de interpolación como los usados por el codificador de vídeo 20 durante la codificación de los bloques de vídeo para calcular valores interpolados para píxeles fraccionarios de bloques de referencia. En este caso, la unidad de compensación de movimiento 72 puede determinar los filtros de interpolación usados por el codificador de vídeo 20 a partir de los elementos de sintaxis recibidos y usar los filtros de interpolación para producir bloques predictivos.

De acuerdo con aspectos de la presente divulgación, el decodificador de vídeo 30 se puede configurar para realizar cualquier combinación de las técnicas descritas en el presente documento para derivar información de movimiento, realizar divisiones de bloques y/o interpolar datos de vídeo. En particular, determinadas técnicas de la presente divulgación se pueden realizar por la unidad de derivación 75. Por ejemplo, de acuerdo con los aspectos de la presente divulgación, la unidad de derivación 75 se puede configurar para determinar la información de movimiento para un bloque actual y sin decodificar la información de movimiento de un flujo de bits codificado.

En algunos casos, la unidad de derivación 75 puede determinar si se realiza la derivación de movimiento para un bloque particular (por ejemplo, frente a la intrapredicción o la interpredicción tradicional). Por ejemplo, el decodificador de vídeo 30 puede decodificar uno o más elementos de sintaxis que indican que se deriva (en lugar de señalizarse) la información de movimiento para un bloque que se está decodificando. El decodificador de vídeo 30 también puede decodificar uno o más elementos de sintaxis que indican un modo de derivación de información de movimiento a partir de una pluralidad de modos de derivación de información de movimiento que se van a usar para decodificar un bloque. La unidad de derivación 75 puede determinar si se realiza la derivación de movimiento y el modo de derivación de información de movimiento para un bloque en base a la sintaxis decodificada. En algunos ejemplos, como se describe en el presente documento, la sintaxis se puede asociar con uno o más de otros modos, tal como el modo de fusión, AMVP u otras funciones de decodificación.

De acuerdo con otros aspectos de la presente divulgación, la unidad de derivación 75 puede interpolar datos de vídeo que no están incluidos en el flujo de bits codificado para una secuencia de vídeo. Por ejemplo, la unidad de



derivación 75 puede realizar cualquier combinación de técnicas de derivación de movimiento para interpolar una imagen que no está incluida en el flujo de bits analizado, por ejemplo, similar a la conversión ascendente de velocidad de tramas. En algunos casos, el descodificador de vídeo 30 puede usar la imagen interpolada durante la codificación. Por ejemplo, la unidad de derivación 75 puede interpolar una imagen y el descodificador de vídeo 30 puede descodificar datos para una imagen actual en relación con la imagen interpolada. En este ejemplo, el descodificador de vídeo 30 puede añadir la imagen interpolada a la memoria de imágenes de referencia 82 y descodificar los datos de otras imágenes en base a al menos una parte de la imagen interpolada.

En otros ejemplos, la unidad de derivación 75 puede interpolar una imagen y el descodificador de vídeo 30 puede establecer la imagen interpolada igual a la imagen actual. Por ejemplo, la unidad de derivación 75 puede interpolar la imagen actual y el descodificador de vídeo 30 puede descodificar elementos de sintaxis para la imagen actual a partir del flujo de bits codificado (por ejemplo, datos de cabecera de fragmento y similares), pero puede omitir la descodificación de datos de vídeo para la imagen actual y en su lugar interpolar la imagen actual.

Si bien la unidad de derivación 75 se puede configurar para realizar determinadas técnicas de interpolación, como se describe en el presente documento, se debe entender que una o más de otras unidades del descodificador de vídeo 30 también o de forma alternativa se pueden configurar para interpolar datos. Por ejemplo, el descodificador de vídeo 30 puede incluir una variedad de otros interpoladores o filtros, por ejemplo, para interpolar un píxel en una ubicación de subpíxel (subpíxel) durante la compensación de movimiento.

La unidad de cuantificación inversa 76 cuantifica de forma inversa, es decir, descuantifica, los coeficientes de transformada cuantificados proporcionados en el flujo de bits y descodificados por la unidad de descodificación entrópica 70. El procedimiento de cuantificación inversa puede incluir el uso de un parámetro de cuantificación  $QP_Y$ , calculado por el descodificador de vídeo 30 para cada bloque de vídeo en el fragmento de vídeo para determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa que se debe aplicar.

La unidad de transformada inversa 78 aplica una transformada inversa, por ejemplo, una DCT inversa, una transformada de enteros inversa o un procedimiento de transformada inversa conceptualmente similar, a los coeficientes de transformada para producir bloques residuales en el dominio de píxel.

Después de que la unidad de compensación de movimiento 72 genera el bloque predictivo para el bloque de vídeo actual en base a los vectores de movimiento y otros elementos de sintaxis, el descodificador de vídeo 30 forma un bloque de vídeo descodificado sumando los bloques residuales de la unidad de transformada inversa 78 a los bloques predictivos correspondientes generados por la unidad de compensación de movimiento 72. El sumador 80 representa el componente o los componentes que realizan esta operación de suma. Si se desea, también se puede aplicar un filtro de eliminación de bloques para filtrar los bloques descodificados para eliminar distorsiones de efecto pixelado. También se pueden usar otros filtros de bucle (en el bucle de codificación o bien después del bucle de codificación) para suavizar las transiciones de píxeles o mejorar de otro modo la calidad de vídeo. Los bloques de vídeo descodificados en una trama o imagen dada se almacenan a continuación en la memoria de imágenes de referencia 82, que almacena imágenes de referencia usadas para la compensación de movimiento posterior. La memoria de imágenes de referencia 82 también almacena vídeo descodificado para su presentación posterior en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la FIG. 1.

Las FIGS. 4A y 4B son diagramas conceptuales que ilustran candidatos vecinos espaciales en HEVC. En algunos ejemplos, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden derivar candidatos de vector de movimiento (MV) espacial a partir del bloque vecino 0, el bloque vecino 1, el bloque vecino 2, el bloque vecino 3 o el bloque vecino 4 para PU0.

En algunos casos, las técnicas para generar los candidatos de MV a partir de los bloques difieren para los modos de fusión y AMVP. La FIG. 4A ilustra un ejemplo para el modo de fusión. Por ejemplo, en HEVC, un codificador de vídeo (por ejemplo, tal como el codificador de vídeo 20 y/o el descodificador de vídeo 30 de las FIGS. 1-3) puede derivar hasta cuatro candidatos de MV espaciales. Los candidatos se pueden incluir en una lista de candidatos que tiene un orden particular. En un ejemplo, el orden para el ejemplo de la FIG. 4A puede ser el bloque vecino 0 (A1), el bloque vecino 1 (B1), el bloque vecino 2 (B0), el bloque vecino 3 (A0) y el bloque vecino 4 (B2).

La FIG. 4B ilustra un ejemplo para el modo AMVP. Por ejemplo, en HEVC, el codificador de vídeo puede dividir los bloques vecinos en dos grupos: el grupo izquierdo que incluye el bloque vecino 0 y el bloque vecino 1, y el grupo superior que incluye el bloque vecino 2, el bloque vecino 3 y el bloque vecino 4. Para cada grupo, el candidato de vector de movimiento potencial asociado con un bloque vecino que se refiere a la misma imagen de referencia que la indicada por el índice de referencia señalado (para el bloque que se está codificando actualmente) puede tener la prioridad más alta para elegirse para formar un candidato final del grupo. Es posible que ninguno de los bloques vecinos contenga un vector de movimiento que apunte a la misma imagen de referencia. Por lo tanto, si no se puede encontrar dicho candidato, el codificador de vídeo puede ajustar a escala el primer candidato disponible para formar el candidato final, por tanto se pueden compensar las diferencias de distancia temporales.

De acuerdo con aspectos de la presente divulgación, los candidatos de vector de movimiento, tales como los vectores de movimiento asociados con los bloques vecinos mostrados en las FIGS. 4A y 4B se pueden usar para derivar un vector de movimiento para un bloque. Por ejemplo, el codificador de vídeo puede generar una lista de candidatos que incluye candidatos de vector de movimiento de los bloques vecinos mostrados en las FIGS. 4A y 4B. En este ejemplo, el codificador de vídeo puede usar uno o más de los candidatos de la lista de candidatos como vector de movimiento inicial en un proceso de derivación de información de movimiento (por ejemplo, coincidencia bilateral, coincidencia de plantillas o similar). El codificador de vídeo puede aplicar uno o más de los candidatos de vector de movimiento en una búsqueda de movimiento de un proceso de derivación de vector de movimiento para identificar datos de referencia. El codificador de vídeo puede seleccionar el candidato a partir de la lista que identifica datos de referencia que coinciden de forma estrecha (por ejemplo, como se describe con respecto a las FIGS. 8-9 a continuación). Por ejemplo, el codificador de vídeo puede realizar una búsqueda de movimiento para un primer conjunto de datos de referencia que corresponde a un segundo conjunto de datos de referencia fuera del bloque actual. El codificador de vídeo puede, en algunos casos, refinar además el candidato, por ejemplo, realizando una búsqueda de movimiento adicional en un área indicada por el candidato seleccionado, para determinar un vector de movimiento derivado usando el proceso de derivación de información de movimiento.

Las FIGS. 5A y 5B son diagramas conceptuales que ilustran la predicción de vector de movimiento temporal en HEVC. Un candidato de predictor de vector de movimiento temporal (TMVP), si está habilitado y disponible, se añade a una lista de candidatos de MV después de los candidatos de vector de movimiento espaciales. En HEVC, el proceso de derivación de vector de movimiento para un candidato de TMVP es el mismo tanto para los modos de fusión como AMVP, sin embargo, el índice de referencia objetivo para el candidato de TMVP en el modo de fusión se establece típicamente en cero.

La FIG. 5A ilustra una ubicación de bloque principal (mostrado como bloque "T") para un candidato de TMVP, que es el bloque inferior derecho fuera de la PU coubicada. La ubicación puede compensar el sesgo de los bloques superiores e izquierdos usados para generar candidatos vecinos espaciales. Sin embargo, si el bloque T está ubicado fuera de la fila de CTB actual o la información de movimiento no está disponible, el bloque se sustituye por un bloque central de la PU, como se ilustra por las flechas discontinuas del bloque T en la FIG. 5A.

La figura 5B ilustra la derivación de un candidato de TMVP 84 para un bloque actual 86 de una imagen actual 88 a partir de una PU coubicada 90 de una imagen coubicada 92, como se indica en el nivel de fragmento (por ejemplo, en una cabecera de fragmento). De manera similar al modo temporal directo en AVC, un vector de movimiento del candidato de TMVP puede estar sujeto a escalado de vector de movimiento, que se realiza para compensar las diferencias de distancia, por ejemplo, distancias temporales entre imágenes. Con respecto al escalado de vector de movimiento, se puede configurar un codificador de vídeo (tal como un codificador de vídeo 20 y/o un descodificador de vídeo 30) para determinar inicialmente que el valor de los vectores de movimiento es proporcional a la distancia de las imágenes en el tiempo de presentación. Un vector de movimiento asocia dos imágenes, la imagen de referencia y la imagen que contiene el vector de movimiento (a saber, la imagen contenedora). Cuando se utiliza un vector de movimiento para predecir el otro vector de movimiento, la distancia de la imagen contenedora y la imagen de referencia se calcula en base a los valores de recuento de orden de imágenes (POC).

Para predecir un vector de movimiento, tanto la imagen contenedora asociada para el vector de movimiento como una imagen de referencia del vector de movimiento pueden ser diferentes. Por lo tanto, el codificador de vídeo puede calcular una nueva distancia en base a los valores de POC, y el codificador de vídeo puede ajustar a escala el vector de movimiento en base a estas dos distancias de POC. Para un candidato vecino espacial, las imágenes contenedoras para los dos vectores de movimiento son las mismas, mientras que las imágenes de referencia son diferentes. En HEVC, el ajuste a escala del vector de movimiento se aplica tanto a TMVP como a AMVP para candidatos vecinos espaciales y temporales.

En algunos ejemplos, se puede configurar un codificador de vídeo para determinar uno o más candidatos de vector de movimiento artificiales. Por ejemplo, si una lista de candidatos de vector de movimiento no está completa, el codificador de vídeo puede generar candidatos de vector de movimiento artificiales e insertar los candidatos de vector de movimiento artificiales al final de la lista hasta que la lista incluye un número predeterminado de entradas. En el modo de fusión, existen dos tipos de candidatos de MV artificiales, incluyendo un candidato combinado derivado solo para fragmentos B y un candidato cero. En algunos casos, el candidato cero se usa solo para AMVP si el tipo combinado no proporciona suficientes candidatos artificiales.

Para cada par de candidatos que ya están en la lista de candidatos y tienen la información de movimiento necesaria, los candidatos de vector de movimiento combinados bidireccionales se derivan por una combinación del vector de movimiento del primer candidato que se refiere a una imagen en la lista 0 y el vector de movimiento de un segundo candidato que se refiere a una imagen en la lista 1.

De acuerdo con aspectos de la presente divulgación, los candidatos de vector de movimiento, tales como el TMVP mostrado en las FIGS. 5A y 5B, se pueden usar para derivar un vector de movimiento para un bloque. Por ejemplo, el codificador de vídeo puede generar una lista de candidatos que incluye un TMVP determinado de acuerdo con

el proceso descrito anteriormente. En este ejemplo, el codificador de vídeo puede usar el TMVP como vector de movimiento inicial en un proceso de derivación de información de movimiento (por ejemplo, coincidencia bilateral, coincidencia de plantillas o similares). El codificador de vídeo puede aplicar el TMVP en un proceso de derivación de vector de movimiento para identificar los datos de referencia. El codificador de vídeo puede seleccionar el TMVP en casos en los que el TMVP identifica datos de referencia que coinciden de forma estrecha (por ejemplo, como se describe con respecto a las FIGS. 8-9 a continuación). El codificador de vídeo puede, en algunos casos, refinar además el TMVP para determinar un vector de movimiento derivado usando el proceso de derivación de información de movimiento.

En algunos ejemplos, el codificador de vídeo puede recortar una lista de candidatos que incluye candidatos de vector de movimiento (tales como los descritos con respecto a las FIGS. 4A-5B). Por ejemplo, en algunos casos, es posible que los candidatos de diferentes bloques sean los mismos, lo que disminuye la eficacia de una lista de candidatos de fusión/AMVP. El código de vídeo puede aplicar un proceso de recorte para resolver este problema. El codificador de vídeo puede comparar un candidato con los otros en la lista de candidatos actual para evitar insertar un candidato idéntico. Para reducir la complejidad, el codificador de vídeo puede aplicar solo números limitados de procesos de recorte en lugar de comparar cada uno potencial con todos los demás existentes.

La FIG. 6 es un diagrama conceptual que ilustra un ejemplo de estimación de movimiento (ME) unilateral en conversión ascendente de velocidad de tramas (FRUC). En particular, la FIG. 6 ilustra una trama actual 100, una trama de referencia 102 y una trama interpolada 104. En algunos casos, un descodificador de vídeo o un dispositivo de posprocesamiento pueden interpolar imágenes en base a una o más imágenes de referencia. El descodificador de vídeo o el dispositivo de posprocesamiento pueden interpolar imágenes para convertir de forma ascendente una velocidad de trama original de un flujo de bits codificado. De forma alternativa, el descodificador de vídeo o el dispositivo de posprocesamiento pueden interpolar imágenes para insertar una o más imágenes que se omitieron por un codificador de vídeo para codificar una secuencia de vídeo a una velocidad de trama reducida. En cualquier caso, el descodificador de vídeo o el dispositivo de posprocesamiento interpola las tramas (tales como la trama interpolada 104) que no están incluidas en un flujo de bits codificado que se ha recibido por el descodificador de vídeo usando imágenes que se han decodificado (tales como la trama actual 100 y la trama de referencia 102). El descodificador de vídeo o el dispositivo de posprocesamiento puede interpolar las imágenes usando cualquiera de una serie de técnicas de interpolación, por ejemplo, usando interpolación de tramas compensadas por movimiento, repetición de tramas o promediado de tramas.

Las técnicas de interpolación de tramas mencionadas anteriormente se implementan típicamente después del bucle. Por ejemplo, un descodificador de vídeo típicamente recibe y descodifica un flujo de bits codificado para generar una representación reconstruida de una secuencia de vídeo que incluye la trama actual 100 y la trama de referencia 102. Después del bucle de descodificación, el descodificador de vídeo u otro dispositivo de posprocesamiento puede interpolar imágenes que se van a incluir con la representación reconstruida, incluyendo la trama interpolada 104. En algunos casos, el proceso de interpolación de imágenes se puede denominar conversión ascendente de velocidad de tramas (FRUC), porque la secuencia de imágenes resultante incluye imágenes adicionales (interpoladas) que no se incluyeron en el flujo de bits codificado.

En consecuencia, la tecnología FRUC se puede usar para generar vídeos de alta velocidad de tramas en base a vídeos de baja velocidad de tramas. Se ha usado FRUC en la industria de visualización. Los ejemplos incluyen H. Liu, R. Xiong, D. Zhao, S. Ma, W. Gao, "Multiple Hypotheses Bayesian Frame Rate Up-Conversion by Adaptive Fusion of Motion-Compensated Interpolations", transacciones del IEEE sobre circuitos y sistemas para tecnología de vídeo, vol. 22, n.º 8, agosto de 2012; W. H. Lee, K. Choi, J. B. Ra, "Frame rate up conversion based on variational image fusion", transacciones del IEEE sobre procesamiento de imágenes, vol. 23, n.º 1, enero de 2014; y U. S. Kim, M. H. Sunwoo, "New frame rate up-conversion algorithms with low computational complexity", transacciones del IEEE sobre circuitos y sistemas para tecnología de vídeo, vol. 24, n.º 3, marzo de 2014.

Los algoritmos de FRUC se pueden dividir en dos tipos. Un tipo de procedimientos interpola tramas intermedias por repetición o promediado de tramas sencillo. Sin embargo, este procedimiento proporciona resultados incorrectos en una imagen que contiene mucho movimiento. El otro tipo de procedimiento, llamado FRUC compensada por movimiento (MC-FRUC), considera el movimiento del objeto cuando genera tramas intermedias y consiste en dos etapas: estimación de movimiento (ME) e interpolación compensada por movimiento (MCI). La ME genera vectores de movimiento (MV), que representan el movimiento del objeto usando vectores, mientras que la MCI usa los MV para generar tramas intermedias.

El algoritmo de coincidencia de bloques (BMA) se usa ampliamente para ME en MC-FRUC ya que es sencillo de implementar. El BMA divide una imagen en bloques y detecta el movimiento de esos bloques, por ejemplo, para determinar si los bloques se corresponden. Dos tipos de ME se usan principalmente para BMA: ME unilateral y ME bilateral

Como se muestra en la FIG. 6, la ME unilateral obtiene los MV buscando el mejor bloque coincidente a partir de la trama de referencia 102 de la trama actual 100. A continuación, se puede ubicar el bloque en la trayectoria de movimiento en la trama interpolada de modo que se logre el MV. Como se muestra en la FIG. 6, tres bloques que

incluyen 106A, 106B y 106C de la trama actual 100, la trama de referencia 102 y la trama interpolada 104, respectivamente, están implicados después de la trayectoria de movimiento. Aunque el bloque 106A en la trama actual 100 pertenece a un bloque codificado, es posible que el mejor bloque coincidente 106B en la trama de referencia 102 no pertenezca por completo a un bloque codificado, y tampoco lo haga el bloque 106C en la trama interpolada 104. En consecuencia, se pueden producir regiones superpuestas de los bloques y regiones sin rellenar (huecos) en la trama interpolada.

Para gestionar las superposiciones, los algoritmos de FRUC sencilla simplemente implican promediar y sobrescribir los píxeles superpuestos. Además, los huecos se cubren por los valores de píxeles de una trama de referencia o una actual. Sin embargo, estos algoritmos dan como resultado distorsiones de efecto pixelado y desenfoque. De ahí que se proponen la segmentación de campo de movimiento, la extrapolación sucesiva usando la transformada discreta de Hartley y la restauración de imagen para gestionar los huecos y superposiciones sin incrementar las distorsiones de efecto pixelado y el desenfoque.

De acuerdo con aspectos de la presente divulgación, un codificador de vídeo (tal como un codificador de vídeo 20 y/o un decodificador de vídeo 30) puede generar una trama interpolada 104 en el bucle de codificación o decodificación usando la técnica de coincidencia unilateral mostrada en la FIG. 6. Por ejemplo, el codificador de vídeo puede usar FRUC a nivel de imagen para interpolar la trama interpolada 104 como predictor de la imagen actual, usando la matriz de píxeles reconstruida. En algunos ejemplos, una imagen interpolada de este tipo se puede considerar como una imagen de referencia o la reconstrucción de la trama actual 100. En otros ejemplos, el codificador de vídeo puede establecer la imagen actual igual a la imagen interpolada. Una imagen de este tipo se puede marcar como una imagen desechable y/o una imagen no de referencia por elementos de sintaxis o procesos de decodificación.

En algunos ejemplos, el codificador de vídeo puede interpolar una imagen actual de modo que un modo FRUC sea el único modo permitido, donde el modo FRUC indica la técnica de coincidencia unilateral mostrada en la FIG. 6 o cualquier otra técnica de interpolación o derivación de información de movimiento descrita en el presente documento. De ahí que en lugar de una señalización de estructura de CU basada en árbol cuaternario, todos los bloques pueden tener el mismo tamaño predefinido o señalizado. En algunos casos, solo se puede permitir un subconjunto de modos de codificación de CU válidos, tales como salto normal, fusión normal, modo FRUC y modo intra. De forma alternativa o adicional, se puede permitir un modo de llenado de huecos para FRUC en una imagen o fragmento de este tipo.

De acuerdo con aspectos de la presente divulgación, se puede introducir un mensaje SEI para indicar qué imágenes o subsecuencia de imágenes están codificadas por FRUC a nivel de imagen. Dichas imágenes se pueden desechar sin afectar la calidad de las otras imágenes. Un mensaje SEI de este tipo puede indicar (o indicar adicionalmente) qué nivel(es) temporal(es) contiene(n) imágenes codificadas con FRUC, o contiene(n) solo imágenes codificadas con FRUC. Dicha información en el mensaje SEI también puede estar presente en otros lugares de la sintaxis de alto nivel, tales como PPS, SPS y VPS.

En algunos ejemplos, un codificador de vídeo puede codificar una parte de una imagen e interpolar los datos de vídeo restantes. Por ejemplo, el codificador de vídeo puede codificar una denominada "sugerencia" para la conversión ascendente de velocidad de tramas de lado de decodificador, que puede permitir que los decodificadores inteligentes o ricos en recursos generen opcionalmente las tramas con FRUC. Por ejemplo, varias regiones clave (tales como las regiones rectangulares) se pueden señalar como una sugerencia para dichas tramas con FRUC. Cuando se recibe la sugerencia y opcionalmente se procesa, el procedimiento de FRUC especificado como parte del decodificador se puede usar en primer lugar para las regiones que no son regiones clave, mientras que las regiones clave se deben procesar además por los medios que es posible que no se especifiquen por el decodificador, tales como los procedimientos de llenado de huecos.

Con respecto al llenado de huecos, de acuerdo con aspectos de la presente divulgación, un codificador de vídeo puede implementar técnicas de llenado de huecos basadas en bloques. Por ejemplo, una técnica de llenado de huecos es usar interbloques vecinos espacialmente para predecir el intrabloque actual en la misma CTU. Por ejemplo, el codificador de vídeo puede codificar/descodificar una CTU dos veces. La primera codificación/descodificación es como habitualmente. En la segunda ronda, solo se codifican/descodifican y se sobrescriben los intrabloques. Para un intrabloque, todos sus interbloques vecinos espacialmente en la misma CTU, incluyendo aquellos en la parte inferior derecha del bloque actual, se marcan como disponibles para la intrapredicción. El procedimiento de llenado de huecos también puede ser a nivel de fragmento, mosaico, imagen, otro en cualquier otro nivel. Otro procedimiento de llenado de huecos puede usar una técnica de restauración de imagen. También se pueden aplicar otras técnicas de llenado de huecos.

La FIG. 7 es un diagrama conceptual que ilustra un ejemplo de estimación de movimiento (ME) bilateral en FRUC. En particular, la FIG. 7 ilustra un bloque interpolado 108 de una trama interpolada 110 que se interpola a partir de un bloque actual 112 de una trama actual 114 y un bloque de referencia 116 de una trama de referencia 118. Como se muestra en la FIG. 7, la ME bilateral es otra solución (en MC-FRUC) que se puede usar para evitar los problemas provocados por superposiciones y huecos mostrados en la FIG. 6. La ME bilateral obtiene los MV que pasan a

través del bloque interpolado 108 usando la simetría temporal entre los bloques 112 y 116 de la trama actual 114 y la trama de referencia 118, respectivamente. Como resultado, no genera superposiciones ni huecos. Puesto que se supone que el bloque actual es un bloque que se está procesando, en un orden determinado, por ejemplo, como en el caso de la codificación de vídeo, una secuencia de dichos bloques cubriría la imagen intermedia completa sin solapamiento. Por ejemplo, en el caso de la codificación de vídeo, los bloques se pueden procesar en el orden de descodificación.

De acuerdo con aspectos de la presente divulgación, se puede aprovechar la estimación de movimiento bilateral mostrada en el ejemplo de la FIG. 7 para derivar información de movimiento. Por ejemplo, como se describe con más detalle con respecto a la FIG. 9 a continuación, un codificador de vídeo (tal como un codificador de vídeo 20 o un descodificador de vídeo 30) puede aplicar coincidencia bilateral como un modo de derivación de información de movimiento para derivar información de movimiento durante la codificación. En la coincidencia bilateral, el codificador de vídeo puede realizar una búsqueda de movimiento para un primer conjunto de datos de referencia en una primera imagen de referencia que corresponde a un segundo conjunto de datos de referencia en una segunda imagen de referencia.

De acuerdo con otros aspectos de la presente divulgación, un codificador de vídeo (tal como un codificador de vídeo 20 y/o un descodificador de vídeo 30) puede generar la trama interpolada en el bucle de codificación o descodificación usando la técnica de coincidencia bilateral mostrada en la FIG. 7. Por ejemplo, el codificador de vídeo puede usar FRUC a nivel de imagen para interpolar la imagen interpolada como un predictor de la imagen actual, usando la matriz de píxeles reconstruida. En algunos ejemplos, una imagen interpolada de este tipo se puede considerar como una imagen de referencia o la reconstrucción de la imagen actual. En otros ejemplos, el codificador de vídeo puede establecer la imagen actual igual a la imagen interpolada. Una imagen de este tipo se puede marcar como una imagen desechable y/o una imagen no de referencia por elementos de sintaxis o procesos de descodificación.

La FIG. 8 es un diagrama conceptual que ilustra un ejemplo de derivación de vector de movimiento de lado de descodificador (DMVD) basado en coincidencia de plantillas. Con los códecs de vídeo avanzados, el porcentaje de bits de la información de movimiento en el flujo de bits aumenta cada vez más. En algunos casos, la DMVD puede reducir el coste de bits de la información de movimiento. La DMVD basada en coincidencia de plantillas puede presentar una mejora en la eficacia de codificación, como se describe, por ejemplo, en S. Kamp, M. Wien, "Decoder-side motion vector derivation for block-based video coding", transacciones del IEEE sobre circuitos y sistemas para tecnología de vídeo, vol. 22, n.º 12, diciembre de 2012.

En el ejemplo de la FIG. 8, una imagen actual 120 incluye un objetivo de predicción 122 (por ejemplo, un bloque que se está codificando actualmente) y una plantilla 124. Las imágenes de referencia 126 incluyen una plantilla coubicada 128, la mejor coincidencia 130 y un vector de desplazamiento 132. Un codificador de vídeo (tal como un codificador de vídeo 20 y/o un descodificador de vídeo 30) puede usar la plantilla 124 para buscar la mejor coincidencia para el objetivo de predicción 122 (por ejemplo, en lugar de usar el propio objetivo de predicción 122, que aún no se ha codificado). Por ejemplo, el codificador de vídeo puede realizar una búsqueda de movimiento para identificar un primer conjunto de datos de referencia (por ejemplo, la mejor coincidencia 130) que corresponde a un segundo conjunto de datos de referencia fuera del objetivo de predicción 122 (por ejemplo, la plantilla 124). Como se indica anteriormente, la correspondencia se puede determinar en base a una cantidad de similitud entre los datos de referencia, y se puede denominar en el presente documento como determinación de una "coincidencia" o de "la mejor coincidencia".

En el ejemplo mostrado, el codificador de vídeo puede identificar la plantilla coubicada 128 en las imágenes de referencia 126. A continuación, el codificador de vídeo puede buscar la mejor coincidencia 130, que incluye valores de píxeles que son similares a la plantilla 124. El codificador de vídeo puede determinar el vector de desplazamiento 132 en base al desplazamiento de la plantilla coubicada 128 y la mejor coincidencia 130 en las imágenes de referencia 126.

Suponiendo que la plantilla 124 y el objetivo de predicción 122 provienen del mismo objeto, se puede usar el vector de movimiento de la plantilla como vector de movimiento del objetivo de predicción. De ahí que en el ejemplo de la FIG. 8, el codificador de vídeo puede aplicar el vector de desplazamiento 132 al objetivo de predicción 122. Puesto que la coincidencia de plantillas se lleva a cabo tanto en un codificador de vídeo como en un descodificador de vídeo, el vector de movimiento se puede derivar en el lado de descodificador para evitar el coste de señalización.

De acuerdo con aspectos de la presente divulgación, el codificador de vídeo puede aplicar la coincidencia de plantillas como un modo de derivación de información de movimiento. Por ejemplo, el codificador de vídeo puede aplicar la coincidencia de plantillas para derivar información de movimiento de un bloque actual ubicando la mejor coincidencia entre la plantilla 124 de la imagen actual y los datos de referencia correspondientes en las imágenes de referencia 126. Si bien el ejemplo de la FIG. 8 ilustra la plantilla 124 como un bloque de datos de vídeo en forma de L, se debe entender que se pueden usar otras plantillas. Por ejemplo, el codificador de vídeo puede usar múltiples bloques como plantilla, por ejemplo, uno o más bloques situados a la izquierda del objetivo de predicción 122 y uno o más bloques situados por encima del objetivo de predicción 122.

De acuerdo con aspectos de la presente divulgación, el codificador de vídeo puede aplicar las técnicas de coincidencia de plantillas mostradas en la FIG. 8 usando uno o más vectores de movimiento de una lista de candidatos de vectores de movimiento. Por ejemplo, el codificador de vídeo se puede configurar para determinar uno o más vectores de movimiento candidatos usando cualquier combinación de técnicas descritas en el presente documento (por ejemplo, candidatos de modo de fusión, candidatos de AMVP, un TMVP o similares). A continuación, el codificador de vídeo se puede configurar para aplicar uno o más de los vectores de movimiento candidatos a la plantilla 124 para ubicar la plantilla coubicada 128 (en este ejemplo, la ubicación de la plantilla coubicada 128 viene dictada por el uno o más vectores de movimiento candidatos y no necesariamente coubicada estrictamente). El codificador de vídeo se puede configurar para determinar de los vectores de movimiento candidatos el que da como resultado la mejor coincidencia entre la plantilla 124 y la plantilla coubicada 128.

De acuerdo con aspectos de la presente divulgación, el codificador de vídeo se puede configurar para refinar el vector de movimiento candidato para derivar información de movimiento para el objetivo de predicción 122. Por ejemplo, el codificador de vídeo puede buscar la mejor coincidencia para la plantilla 124 en una región de imágenes de referencia 126 identificadas por el vector de movimiento candidato. Tras la determinación de la mejor coincidencia, el codificador de vídeo puede determinar un desplazamiento entre la plantilla 124 y la coincidencia basada determinada. El codificador de vídeo puede designar el desplazamiento como un vector de movimiento derivado para el objetivo de predicción 122.

La FIG. 9 es un diagrama conceptual que ilustra un ejemplo de derivación de vector de movimiento bidireccional en DMVD. Otra categoría de DMVD es la derivación de MV bidireccional con base especular, como se describe, por ejemplo, en Y.-J. Chiu, L. Xu, W. Zhang, H. Jiang, "Decoder-side Motion Estimation and Wiener filter for HEVC", Comunicaciones visuales y procesamiento de imágenes (VCIP), 2013. El concepto de derivación de vector de movimiento bidireccional en DMVD puede ser similar a la ME bilateral en FRUC. Por ejemplo, la derivación de MV con base especular se puede aplicar por la estimación de movimiento centrosimétrica alrededor de los centros de búsqueda con precisión de muestra fraccionada.

El ejemplo de la FIG. 9 incluye la imagen actual 140 que tiene el bloque actual 142 (el bloque que se está codificando actualmente), un primer vector de movimiento candidato PMV0 que identifica un primer bloque de plantilla 144 de una primera imagen de referencia 146 (L0 ref), y un segundo vector de movimiento candidato PMV1 que identifica un segundo bloque de plantilla 148 de una segunda imagen de referencia 150. El codificador de vídeo puede aplicar dMV como una desviación para ubicar un primer bloque de referencia 152 en la ventana de búsqueda 154 de la primera imagen de referencia 146 y para ubicar un segundo bloque de referencia 156 en la ventana de búsqueda 158 de la segunda imagen de referencia 150.

Por ejemplo, el codificador de vídeo puede añadir dMV a PMV0 y restar dMV de PMV1 para generar un par de MV, MV0 y MV1. El codificador de vídeo puede verificar todos los valores de dMV en el interior de la ventana de búsqueda 154 y 158 para determinar qué valor de dMV da como resultado la mejor coincidencia entre el primer bloque de referencia 152 (por ejemplo, un primer conjunto de datos de referencia) de LO ref y el segundo bloque de referencia 156 (por ejemplo, un segundo conjunto de datos de referencia) de LI ref. En algunos ejemplos, el codificador de vídeo puede determinar la mejor coincidencia en base a la suma de diferencias absolutas (SAD). En otros ejemplos, el codificador de vídeo puede usar otra métrica para determinar la mejor coincidencia. El tamaño y la ubicación de las ventanas de búsqueda 154 y 158 pueden estar predefinidos o se pueden señalar en un flujo de bits.

El codificador de vídeo puede seleccionar el par de MV con la mínima SAD como salida de la estimación de movimiento centrosimétrica. Puesto que esta técnica usa una referencia futura (referencia en una posición temporal posterior a la trama actual) y una referencia anterior (referencia en una posición temporal anterior a la trama actual) para la coincidencia por SAD, no se puede aplicar a la trama P o a las tramas B de bajo retardo en las que solo está disponible la referencia anterior.

De acuerdo con aspectos de la presente divulgación, el codificador de vídeo puede aplicar las técnicas de derivación de vector de movimiento bidireccional como un modo de derivación de información de movimiento. En algunos ejemplos, el codificador de vídeo puede aplicar las técnicas mostradas en la FIG. 9 usando uno o más vectores de movimiento de una lista de candidatos de vectores de movimiento. Por ejemplo, el codificador de vídeo se puede configurar para determinar uno o más vectores de movimiento candidatos usando cualquier combinación de técnicas descritas en el presente documento (por ejemplo, candidatos de modo de fusión, candidatos de AMVP, un TMVP o similares). A continuación, el codificador de vídeo se puede configurar para aplicar uno o más de los vectores de movimiento candidatos como PMV0 y/o PMV1 para ubicar el primer bloque de plantilla 144 y el segundo bloque de plantilla 148. El codificador de vídeo se puede configurar para determinar de los vectores de movimiento candidatos el que da como resultado la mejor coincidencia entre el primer bloque de plantilla 144 y el segundo bloque de plantilla 148.

De acuerdo con aspectos de la presente divulgación, a continuación el codificador de vídeo se puede configurar para refinar el vector de movimiento candidato para derivar información de movimiento para el bloque actual 142.

Por ejemplo, el codificador de vídeo puede buscar la mejor coincidencia aplicando una variedad de valores de dMV, de la manera descrita anteriormente. De esta forma, el codificador de vídeo puede derivar el par de MV MV0 y MV1.

La FIG. 11 es un diagrama conceptual que ilustra la derivación de vector de movimiento basada en coincidencia bilateral extendida. Un inconveniente potencial de la derivación de MV bidireccional con base especular (por ejemplo, como se muestra en la FIG. 10) es que no funciona cuando dos referencias de la imagen actual son ambas anteriores o ambas posteriores a la imagen actual. Las técnicas de coincidencia bilateral extendida descritas en el presente documento pueden, en algunos casos, superar el inconveniente de que todas las imágenes de referencia de la imagen actual están en el mismo lado (en orden de visualización) que la imagen actual.

El ejemplo de la FIG. 11 incluye una imagen actual 160 que incluye un bloque actual 162, una primera imagen de referencia (Ref0) 164 que incluye un primer bloque de referencia 166 y una segunda imagen de referencia (Refl) 168 que incluye un segundo bloque de referencia 170. Como se muestra en la FIG. 11, la primera imagen de referencia (Ref0) 164 y la segunda imagen de referencia (Refl) 168 están ubicadas ambas antes de la imagen actual en la dirección temporal. Suponiendo que el primer bloque de referencia 166, el segundo bloque de referencia 170 y el bloque actual 162 están a lo largo de la misma trayectoria de movimiento, la proporción entre MV0 y MV1 será igual a la proporción entre la distancia temporal TDO y TD1. En otras palabras, dadas dos referencias Ref0 y Refl con distancia temporal TDO y TD1 con respecto a la imagen actual, para cualquier MV0 en Ref0, se puede determinar MV1 en Refl ajustando a escala MV0.

El codificador de vídeo puede seleccionar el par MV0 y MV1 final como el par que minimiza el coste de coincidencia entre el par de bloques apuntado por MV0 y MV1 (por ejemplo, como se describe anteriormente con respecto a la FIG. 10). Teóricamente, el bloque actual 162 se puede considerar como un bloque extrapolado en base al primer bloque de referencia 166 y el segundo bloque de referencia 170. Cabe destacar que la coincidencia bilateral extendida también funciona en el caso bidireccional habitual en el que la imagen actual está temporalmente entre las dos referencias. En este caso, el bloque actual 162 se puede considerar como un bloque interpolado en base al primer bloque de referencia 166 y el segundo bloque de referencia 170. Además, las técnicas de coincidencia bilateral descritas en el presente documento no requieren una "relación especular" entre MV0 y MV1, incluso en el caso bidireccional. El supuesto de coincidencia bilateral es que la proporción entre MV0 y MV1 es proporcional a la proporción entre la distancia temporal de Ref0 con respecto a la imagen actual y la de Refl con respecto a la imagen actual.

Claramente, para bloques de referencia distintos del primer bloque de referencia 166 y el segundo bloque de referencia 170, el codificador de vídeo puede derivar un par de MV diferente. En un ejemplo, el descodificador de vídeo puede seleccionar imágenes de referencia para realizar una correspondencia bilateral de acuerdo con un orden en el que aparecen las imágenes de referencia en una lista de imágenes de referencia. Por ejemplo, el codificador de vídeo puede seleccionar la primera referencia en la lista de referencias 0 como Ref0 y la primera referencia en la lista de referencias 1 como Refl. A continuación, el codificador de vídeo puede buscar el par de MV (MV0, MV1). En otro ejemplo, el codificador de vídeo selecciona Ref0 en base a una entrada en una lista inicial (por ejemplo, una lista de candidatos de vector de movimiento inicial). A continuación, el codificador de vídeo puede establecer Refl en una imagen de referencia en la otra lista de imágenes de referencia que esté temporalmente lo más cercana a la imagen actual. En consecuencia, el codificador de vídeo puede buscar el par de MV (MV0, MV1) en Ref0 y Refl.

De ahí, de acuerdo con aspectos de la presente divulgación, el codificador de vídeo puede aplicar las técnicas de derivación de movimiento bidireccional extendida ilustradas en la FIG. 11 como un modo de derivación de información de movimiento. Por ejemplo, el codificador de vídeo puede usar la coincidencia bilateral para derivar información de movimiento del bloque actual 162 encontrando la mejor coincidencia entre dos bloques (por ejemplo, tal como el primer bloque de referencia 166 y el segundo bloque de referencia 170) a lo largo de la trayectoria de movimiento del bloque actual en dos imágenes de referencia diferentes. En el supuesto de trayectoria de movimiento continuo, los vectores de movimiento MV0 y MV1 que apuntan a los dos bloques de referencia, primer bloque de referencia 166 y segundo bloque de referencia 170, serán proporcionales a las distancias temporales, es decir, TDO y TD1, entre la imagen actual y las dos imágenes de referencia. Como caso especial, cuando la imagen actual 160 está temporalmente entre dos imágenes de referencia (como se muestra en el ejemplo de la FIG. 10) y la distancia temporal desde la imagen actual a las dos imágenes de referencia es la misma, la coincidencia bilateral se vuelve MV bidireccional con base especular.

La FIG. 11 es un diagrama de flujo que ilustra un ejemplo de descodificación de una unidad de predicción (PU) usando DMVD. En Y.-J. Chiu, L. Xu, W. Zhang, H. Jiang, "Decoder-side Motion Estimation and Wiener filter for HEVC", Comunicaciones visuales y procesamiento de imágenes (VCIP), 2013, se propuso además combinar la derivación de MV bidireccional con base especular con el modo de fusión en HEVC. En la técnica propuesta, se añade un indicador llamado pu\_dmvd\_flag para una PU de fragmentos B para indicar si se aplica un modo DMVD a la PU actual. Debido a que el modo DMVD no transmite explícitamente ninguna información de MV en el flujo de bits, el elemento de sintaxis pu\_dmvdflag se integra con la sintaxis del modo de fusión en HEVC (que usa un índice para los datos representativos de un vector de movimiento en lugar del propio vector de movimiento).

En el ejemplo de la FIG. 11, un decodificador de vídeo (tal como un decodificador de vídeo 30) puede comenzar a decodificar una PU (180). El decodificador de vídeo 30 puede determinar si el modo usado para decodificar la PU es el modo de fusión (182), por ejemplo, en base a la sintaxis incluida en un flujo de bits que incluye la PU.

5 Si no se usa el modo de fusión (la rama "no" de la etapa 182), el decodificador de vídeo 30 puede usar un proceso normal para una PU no de fusión para decodificar la PU (184) y finalizar el proceso (186).

Si se usa el modo de fusión (la rama "sí" de la etapa 182), el decodificador de vídeo 30 puede determinar si se usa DMVD para determinar la información de movimiento para la PU en base al valor del elemento de sintaxis `pu_dmvd_flag` (188). Si no se usa DMVD (la rama "no" de la etapa 188), el decodificador de vídeo 30 puede usar un modo de fusión normal para decodificar la PU (190) y finalizar el proceso (186). Si se usa DMVD (la rama "sí" de la etapa 188), el decodificador de vídeo 30 puede aplicar un proceso de DMVD para determinar la información de movimiento para la PU (192) y finalizar el proceso (186).

15 En algunos casos, las técnicas de codificación de vídeo actuales pueden tener determinadas limitaciones. Por ejemplo, determinadas técnicas de DMVD se pueden considerar como un subconjunto de la tecnología FRUC. Aunque se ha aplicado DMVD en la codificación de vídeo, no se han implementado otras técnicas de FRUC eficaces en la codificación de vídeo, por ejemplo, en el bucle de codificación de vídeo por un codificador de vídeo o un decodificador de vídeo. Además, aunque se han propuesto diferentes técnicas de DMVD, la interacción y superposición de dichas técnicas no se aprovechan en el decodificador. Es decir, solo se ha usado un modo

20 DMVD en otros sistemas de codificación de vídeo. No se ha estudiado la manera en la que usar una pluralidad de técnicas de DMVD para mejorar además la eficacia de codificación.

Como otra limitación potencial, DMVD se puede aplicar solo a bloques relativamente grandes de datos de vídeo y, por lo tanto, es posible que no sea muy eficaz. La aplicación de dichos procedimientos para bloques más pequeños puede dar lugar a una sobrecarga apreciable debido al coste de señalización. En algunos casos, el intervalo de búsqueda para las técnicas de DMVD tradicionales puede ser relativamente pequeño y solo se buscan varios puntos, por ejemplo, 16 puntos. Además, como se indica anteriormente, la derivación de MV bidireccional con base

25 especular no se puede aplicar en el caso de B de bajo retardo, porque es necesario identificar dos imágenes de referencia con orden de visualización antes y después de una imagen actual y esto no es posible en el caso de bajo retardo.

Otra limitación potencial puede ser que, en el decodificador, el coste de coincidencia de las técnicas de DMVD tradicionales solo puede considerar la distorsión. Sin embargo, la magnitud del vector de movimiento no se ha considerado en el coste de coincidencia, lo que puede dar lugar a una optimización local o un resultado inexacto de la coincidencia, por ejemplo, debido al ruido en el dominio de píxeles. Además, la complejidad de las técnicas de DMVD tradicionales puede ser relativamente alta, tanto en términos de ancho de banda de memoria como de complejidad informática, especialmente debido al hecho de que la interpolación es necesaria para los vectores de movimiento de píxeles fraccionarios durante la búsqueda en el decodificador.

Las técnicas de la presente divulgación pueden abordar una o más de las limitaciones potenciales descritas anteriormente. En algunos ejemplos, las técnicas para derivar información de movimiento se pueden aplicar individualmente. De forma alternativa, cualquier combinación de las técnicas descritas en el presente documento se pueden aplicar conjuntamente. Como se describe en el presente documento, la información de índice de

40 referencia se puede considerar, en general, como parte de la información de movimiento. En algunos casos, la información de vector de movimiento y la información de índice de referencia se pueden denominar conjunto de información de movimiento.

Determinadas técnicas de la presente divulgación incluyen seleccionar un modo de derivación de movimiento de una pluralidad de modos de derivación de movimiento potenciales. Por ejemplo, de acuerdo con los aspectos de la presente divulgación, un codificador de vídeo (por ejemplo, un codificador de vídeo 20 o un decodificador de vídeo 30) puede usar dos o más técnicas de derivación de movimiento para proporcionar una mejor predicción de la información de movimiento en un decodificador de vídeo para evitar la señalización de la información de movimiento en el flujo de bits. Las dos o más técnicas de derivación de movimiento pueden incluir, pero no se

50 limitan a, coincidencia bilateral, coincidencia de plantillas y cualquier otro procedimiento de coincidencia. Estas técnicas se pueden denominar, en general, modos de derivación de información de movimiento o modos FRUC. De ahí se debe entender que en algunos casos se puede usar en cambio una técnica denominada modo FRUC para interpolar información de movimiento para un bloque actual (por ejemplo, en lugar de interpolar un nuevo bloque de datos de vídeo).

En algunos ejemplos, cuando se usan múltiples procedimientos de derivación de movimiento, en lugar de tener diferentes procedimientos de optimización para diferentes procedimientos de derivación, el proceso para encontrar el mejor movimiento para dos o más de los procedimientos de derivación de movimiento puede estar alineado, en términos de la selección de los puntos de partida para la búsqueda y cómo buscar alrededor de los puntos de

60 partida. Por ejemplo, el codificador de vídeo puede construir una lista de candidatos de vector de movimiento, seleccionar un candidato inicial de la lista y refinar el candidato usando las mismas técnicas de búsqueda y



coincidencia. En este ejemplo, la derivación de movimiento bidireccional y la derivación de movimiento basada en coincidencia de plantillas se pueden usar de una manera adaptativa en el lado de descodificador.

De acuerdo con aspectos de la presente divulgación, se introduce señalización adicional a nivel de bloque para identificar qué procedimiento de derivación de movimiento se usa para codificar el bloque actual. Por ejemplo, el codificador de vídeo puede codificar uno o más elementos de sintaxis para indicar si está habilitada la derivación de información de movimiento. El codificador de vídeo también puede codificar uno o más elementos de sintaxis para indicar un modo de derivación de información de movimiento particular a partir de una pluralidad de modos potenciales. En otros ejemplos, es posible que la técnica de derivación de información de movimiento que se va a usar no se señalice, sino que se derive en el descodificador de vídeo, por ejemplo, en base a un modo de predicción u otra información disponible antes de que se decodifique el bloque actual. Todavía en otros ejemplos, el codificador de vídeo puede realizar múltiples modos de derivación de información de movimiento y determinar un promedio ponderado de las predicciones de las dos o más técnicas de derivación para codificar el bloque actual.

En un ejemplo con propósitos ilustrativos, un descodificador de vídeo, tal como un descodificador de vídeo 30, puede identificar en primer lugar si se aplica un modo de interpolación de información de movimiento (por ejemplo, en base a la sintaxis en el flujo de bits). A continuación, el descodificador de vídeo 30 puede identificar qué modo de derivación de información de movimiento se va a aplicar (por ejemplo, en base a la sintaxis en el flujo de bits). En un ejemplo, cuando se admiten múltiples procedimientos de derivación de movimiento simultáneamente, por ejemplo, ambos modos de coincidencia bilateral extendida y coincidencia de plantillas, se puede señalar un valor de índice en el flujo de bits para indicar qué procedimiento de derivación de movimiento está realmente en uso para un bloque actual. El índice puede tener tres valores que incluyen desactivado, coincidencia bilateral extendida y coincidencia de plantillas.

Cuando se usa la codificación CABAC, se pueden usar dos intervalos (*bins*) para representar el índice. Ambos de los dos intervalos pueden usar vecinos espaciales como contextos o solo uno de ellos usa contextos de vecinos espaciales. De forma alternativa, uno o ambos intervalos pueden usar otra información codificada, tal como la profundidad de CU, como contextos. La binarización del índice puede estar predefinida, tal como "0" para desactivado, "10" para coincidencia bilateral extendida y "11" para coincidencia de plantillas. De forma alternativa, la binarización del índice se puede señalar en una cabecera de fragmento o derivarse de información codificada tal como tipo de fragmento, nivel temporal o información QP.

De acuerdo con aspectos de la presente divulgación, la sintaxis que indica un modo de derivación de información de movimiento se puede incluir con otro modo de codificación. En un ejemplo, un modo de derivación de información de movimiento se puede considerar un candidato de fusión particular, indicado, por tanto, por un índice de fusión. En este caso, el tamaño de la lista de candidatos de fusión se puede incrementar para adaptar el índice adicional. El índice de candidatos de fusión para el modo de derivación de información de movimiento puede estar predefinido o señalizarse en el flujo de bits.

En algunos ejemplos, tanto la coincidencia bilateral extendida como la coincidencia de plantillas se admiten con el modo de fusión. En dichos ejemplos, cuando el indicador de fusión es igual a 1, se señala un nuevo indicador de derivación de información de movimiento para indicar si la información de movimiento se deriva para la PU actual. El indicador puede usar el mismo indicador de su vecino espacial, tal como los bloques superior e izquierdo como contextos de codificación CABAC. Cuando este indicador está activado, se señala un segundo indicador para indicar qué modo de derivación de información de movimiento (por ejemplo, coincidencia bilateral extendida o coincidencia de plantillas) se usa para derivar la información de movimiento del bloque. Cuando la derivación de movimiento está activada, incluso cuando el modo actual es el modo de fusión, no se señala ningún índice de fusión. De forma alternativa o adicional, no se permite una derivación de información de movimiento particular (por ejemplo, tal como la coincidencia de plantillas) si la PU no es la primera PU de una CU en el orden de descodificación. En este caso, solo es necesario señalar el indicador para indicar si se usa o no la concordancia bilateral extendida para la PU.

En otro ejemplo, el modo de derivación de información de movimiento se puede combinar con el modo AMVP, por ejemplo, al nivel de PU. En un ejemplo, se pueden señalar elementos de sintaxis adicionalmente (por ejemplo, un indicador adicional) para indicar el modo de derivación de información de movimiento. Cuando este modo está activado, no se puede señalar ningún índice de AMVP en el flujo de bits. De otro modo, se puede señalar un índice de AMVP normal en el flujo de bits. En otro ejemplo, el modo de derivación de información de movimiento se puede considerar como un candidato de AMVP particular, tal como el primer candidato de AMVP, en una lista de candidatos de AMVP. En algunos casos, cuando se combina con el modo AMVP, es posible que un vector de movimiento no se derive a un nivel de sub-PU.

De acuerdo con otros aspectos, un codificador de vídeo puede inicialmente codificar condicionalmente un indicador de nivel de CU (por ejemplo, para una CU intercodificada) para indicar si todas las PU dentro de la CU actual usan el modo de derivación de información de movimiento. En un ejemplo, el indicador de nivel de PU no se señala. En otro ejemplo, cuando el indicador de CU es igual a 0 (es decir, no todas las PU están codificadas con el modo),

el indicador de nivel de PU de la primera PU se señala además mientras que la segunda PU no incluye el indicador de nivel de PU.

En algunos ejemplos, el modo de derivación de información de movimiento se puede deshabilitar para tipos de fragmentos, niveles temporales, tipos de bloques o tamaños de bloques específicos. En un ejemplo, no se permite la derivación de información de movimiento cuando el fragmento actual solo incluye imágenes de referencia con posiciones temporales que están todas antes o después de la de la imagen actual. En otro ejemplo, la derivación de información de movimiento no está permitida para PU que no sean  $2N \times 2N$ . Cuando se deshabilita la derivación de información de movimiento, no se necesita señalización a nivel de bloque relacionada con la derivación de información de movimiento.

De acuerdo con aspectos de la presente divulgación, habilitar o deshabilitar las técnicas de derivación de información de movimiento descritas en el presente documento se puede controlar por una sintaxis de alto nivel para proporcionar una mejor compensación de complejidad frente a eficacia de codificación y/o flexibilidad de codificador/descodificador. En un ejemplo, se puede señalar un indicador en una SPS, PPS, cabecera de fragmento o cualquier otra cabecera de sintaxis de alto nivel para indicar el uso de un modo de derivación de información de movimiento. Cuando este indicador indica que este modo de codificación no está habilitado, es posible que no se codifiquen los indicadores de nivel de CU/PU en el flujo de bits.

En algunos casos, la sintaxis de alto nivel se puede usar de forma adicional o alternativa para indicar otros parámetros de derivación de información de movimiento. Por ejemplo, un índice del algoritmo de búsqueda que se va a usar para buscar el vector de movimiento a nivel de PU se puede codificar en un flujo de bits dentro de una SPS, PPS o cabecera de fragmento. En algunos casos, un índice del algoritmo de búsqueda que se usa para buscar el vector de movimiento a nivel de subbloque se puede codificar en el flujo de bits dentro de una SPS, PPS o cabecera de fragmento. En algunos ejemplos, para mantener una baja complejidad informática en el lado de descodificador, los números máximos de bloques/coincidencias de división en el nivel de PU, los números máximos de bloques/coincidencias de división en el nivel de sub-PU y/o el número de coincidencias total de en tanto el nivel de PU como el de sub-PU pueden estar restringidos. En algunos casos, dichos números máximos pueden estar predefinidos o señalizarse en el flujo de bits.

Las técnicas de la presente divulgación incluyen una variedad de técnicas para derivar información de movimiento. En algunos ejemplos, el codificador de vídeo puede determinar una lista inicial (lista de candidatos iniciales) de entradas que contienen vectores de movimiento, y un mejor punto de partida se identifica como una entrada de la lista inicial. Las entradas que contienen vectores de movimiento pueden ser vectores de movimiento de vecinos espaciales, vecinos temporales y/o vectores de movimiento derivados por otros medios. De forma alternativa, el mejor punto de partida (o índice del mejor punto de partida) se puede señalar al descodificador.

En algunos ejemplos, la lista inicial puede contener los vectores de movimiento de los vecinos espaciales y/o temporales. Cada entrada de la lista inicial puede ser un conjunto unipredictivo de información de movimiento, que incluye un vector de movimiento y su índice de referencia. En un ejemplo, la lista inicial se puede generar de la misma manera que la lista de candidatos de predicción de movimiento usada en otro modo de codificación, por ejemplo, la misma que la lista de candidatos de fusión. En este caso, se pueden usar hasta dos vectores de movimiento de cada candidato de fusión para generar hasta dos entradas en la lista inicial. En algunos casos, las entradas en la lista inicial se pueden generar a partir de un subconjunto de candidatos de predicción de movimiento en una lista usada en otro modo de codificación, por ejemplo, un subconjunto de la lista de candidatos de fusión.

En otro ejemplo, se pueden añadir vectores de movimiento adicionales a la lista inicial, además de los de la lista de candidatos de predicción de movimiento usados en otro modo de codificación, por ejemplo, además de los de la lista de candidatos de fusión. De forma alternativa o adicional, los vectores de movimiento de los vecinos espaciales del bloque actual, tales como un bloque superior, un bloque izquierdo, un bloque superior derecho u otro bloque se pueden añadir a la lista inicial. En algunos casos, también se pueden añadir a la lista vectores de movimiento cero con diferentes índices de imagen de referencia.

De forma alternativa o adicional, se pueden añadir a la lista inicial los vectores de movimiento de los bloques coubicados temporalmente del bloque actual (por ejemplo, un TMVP para el bloque actual) y/o vectores de movimiento de los bloques coubicados temporalmente en la parte inferior derecha del bloque actual en las imágenes de referencia. Antes de añadir un vector de movimiento candidato particular a la lista, los vectores de movimiento se pueden ajustar a escala o no en base a la distancia temporal.

De acuerdo con algunos aspectos, un vector de movimiento se puede interpolar y/o extrapolar a partir de una imagen de referencia y se puede añadir en la lista inicial. Por ejemplo, antes de codificar una interimagen, se puede generar un campo de movimiento interpolado y/o extrapolado de la imagen en base a sus imágenes de referencia con una técnica similar a ME unilateral. El campo de movimiento interpolado y/o extrapolado de un bloque se puede usar para la predicción de MV o usarse como candidatos de partida adicionales en una búsqueda de MV de un modo de derivación de información de movimiento. Téngase en cuenta que el campo de movimiento interpolado y/o extrapolado se guarda en la unidad de  $4 \times 4$  u  $8 \times 8$  o cualquier otro nivel de bloque predefinido/señalizado, y una

PU puede contener múltiples bloques de este tipo de modo que se puedan usar múltiples MV interpolados y/o extrapolados.

En un ejemplo, el campo de movimiento de cada imagen de referencia en ambas listas de referencia atraviesa el bloque de NxN por el bloque de NxN, donde N puede estar predefinido como 4, 8 o 16, o señalizado. Para cada bloque, si el movimiento asociado al bloque que pasa a través de un bloque de NxN en la imagen actual y al bloque no se le ha asignado ningún movimiento interpolado, el movimiento del bloque de referencia se ajusta a escala a la imagen actual de la misma manera que el del escalado de MV en TMVP y el movimiento ajustado a escala se asigna al bloque en la trama actual. Si no se asigna ningún MV ajustado a escala a un bloque de NxN, el movimiento del bloque se marca como no disponible en el campo de movimiento interpolado. En otro ejemplo, a un bloque de NxN en la imagen actual se le pueden asignar múltiples vectores de movimiento en el campo de movimiento interpolado.

En algunos casos, un codificador de vídeo puede recortar uno o más candidatos de una lista de candidatos. Se puede aplicar recortado para eliminar entradas idénticas de una lista inicial antes del mejor proceso de selección de punto de partida, por ejemplo, antes de calcular el coste de coincidencia para cada candidato de la lista.

En algunos casos, la primera imagen en cada lista de referencia se puede usar como imagen de referencia, y los candidatos de vector de movimiento se ajustan a escala en consecuencia si es necesario. En un caso de este tipo, se puede modificar el índice de referencia de cada entrada de la lista inicial después de ajustar a escala el vector de movimiento en base, por ejemplo, a la distancia de POC, de forma similar a TMVP. En algunos casos, el índice de referencia de cada entrada se puede fijar en una o dos imágenes y los vectores de movimiento asociados se pueden ajustar a escala hacia dichas imágenes.

En un ejemplo, para la coincidencia bilateral, se puede obtener un par de vectores de movimiento, que es un conjunto completo de información de movimiento que contiene tanto los vectores de movimiento como sus índices de referencia asociados a la lista 0 y la lista 1 de referencia, respectivamente, en base a cada entrada de una lista de candidatos inicial. A continuación, un codificador de vídeo puede realizar una coincidencia bilateral para todos los pares de MV generados a partir de todas las entradas en la lista inicial, y seleccionar el candidato que da lugar al coste de coincidencia mínimo.

Para generar el par de MV, el codificador de vídeo puede seleccionar una entrada de la lista inicial como primer vector de movimiento y generar el segundo vector de movimiento. Por ejemplo, supóngase que una entrada contiene el primer vector de movimiento  $MV_A$  actual y un índice de referencia (con el valor de POC  $POC_A$ ) está asociado a una primera imagen de referencia en la lista A de imágenes de referencia (siendo A igual a 0 o 1). Suponiendo que el valor de POC de la imagen actual es  $POC_C$ , el codificador de vídeo se puede configurar para encontrar una segunda imagen de referencia a partir de la lista B de imágenes de referencia (siendo B igual a  $1-A$ ) de modo que su valor de POC  $POC_B$  sea igual a  $(2 \times POC_C - POC_A)$ . Si ninguna imagen de referencia en la lista B de imágenes de referencia tiene un valor de POC igual a  $(2 \times POC_C - POC_A)$ , el codificador de vídeo puede seleccionar la segunda imagen de referencia verificando todas las imágenes de referencia en la lista B de modo que  $POC_B$  no sea igual a  $POC_A$  y el valor absoluto de  $POC_C - POC_B$  sea el mínimo. En resumen, el codificador de vídeo puede seleccionar una imagen ubicada en el otro lado de la imagen actual (en orden de visualización) que tenga la misma distancia de POC. Si no está disponible, el codificador de vídeo puede seleccionar una imagen en el otro lado que tenga la distancia más pequeña a la imagen actual. Si todas las imágenes de referencia están en el mismo lado temporal que la primera referencia con  $POC_A$  cuando se compara con la imagen actual, el codificador de vídeo puede seleccionar la referencia que está temporalmente lo más cercana a la imagen actual y tiene un POC distinto de  $POC_A$ . En los supuestos anteriores, el codificador de vídeo puede ajustar a escala el primer vector de movimiento  $MV_A$  para generar el segundo vector de movimiento asociado con la segunda imagen de referencia, por ejemplo, en base a la diferencia de POC como en TMVP. De forma alternativa, se puede usar cualquier técnica en base a la suposición de MV bilateral para generar la segunda imagen de referencia y su vector de movimiento.

De acuerdo con algunos aspectos, se pueden seleccionar dos vectores de movimiento de partida de la lista inicial. Cada uno de estos dos vectores de movimiento se selecciona (en términos de coste de coincidencia mínimo) de las entradas en dos subconjuntos de la lista inicial. Cada sub subconjunto contiene vectores de movimiento asociados con el índice de referencia solo a la lista de imágenes de referencia 0 o solo a la lista de imágenes de referencia 1.

De acuerdo con los aspectos de la presente divulgación, un codificador de vídeo se puede configurar para seleccionar un candidato a partir de un candidato en base a un coste de coincidencia asociado con el vector de movimiento. Además, después de seleccionar un candidato de una lista de candidatos, el codificador de vídeo puede refinar el vector de movimiento candidato para derivar información de movimiento para un bloque. Por ejemplo, el vector de movimiento candidato se puede usar para indicar un punto de partida de una imagen de referencia, que a continuación se puede buscar para encontrar la mejor coincidencia en base a un coste de coincidencia.

De acuerdo con los aspectos de la presente divulgación, se puede usar una variedad de costes de coincidencia, por ejemplo, cuando se determina la mejor coincidencia para un candidato de vector de movimiento inicial y/o cuando se refina el candidato de vector de movimiento inicial. En un ejemplo, cuando se busca el movimiento de un bloque, se puede incluir un coste de MV en el coste de coincidencia para evitar un impacto negativo por el ruido.

Por ejemplo, se puede usar la parte de refinamiento del vector de movimiento (por ejemplo, la diferencia entre el MV actual y el centro de búsqueda), indicado como MVR para calcular el coste. En este ejemplo, el coste puede ser  $w \cdot (|MVR[0]| + |MVR[1]|)$ , donde  $w$  es un factor de ponderación que se puede señalar o predefinir y MVR[0] y MVR[1] son los dos componentes de MVR. De forma alternativa, se puede usar el vector de movimiento refinado MV para calcular el coste, por ejemplo, como  $w \cdot (|MV[0]| + |MV[1]|)$ .

En algunos ejemplos, cuando el bloque es relativamente pequeño, por ejemplo, de 4x4 o más pequeño, se puede usar un bloque más grande que cubra el bloque (por ejemplo, el bloque con límites extendidos) en el cálculo del coste de coincidencia para suprimir el ruido. Por ejemplo, cuando se busca la mejor coincidencia para un bloque de 4x4, el coste de coincidencia se puede calcular en base al bloque de 8x8 siendo el bloque un bloque central.

En algunos ejemplos, el coste de coincidencia puede ser cualquier tipo de distancia/coste, tal como la suma de diferencias absolutas (SAD), la suma de errores de predicción al cuadrado (SSE) o la suma de diferencias absolutas transformadas (SATD). Para reducir la complejidad informática, la SAD, la SSE u otro coste se puede calcular con resolución reducida en dirección horizontal, vertical o en ambas direcciones. Por ejemplo, para un bloque de 8x8, la SAD se puede calcular en base solo a filas impares. En otro ejemplo, el coste de coincidencia se puede calcular en base a un subconjunto seleccionado de un bloque, por ejemplo, solo se puede usar una región central de un.

De acuerdo con los aspectos de la presente divulgación, el proceso de refinamiento para la mejor coincidencia (por ejemplo, el candidato seleccionado en base al coste de coincidencia) se puede realizar dentro de una ventana de búsqueda predefinida o señalizada en lugar de usar siempre una ventana pequeña, por ejemplo, dentro de una ventana de 2x2 (aquí la unidad es el píxel y los vectores de movimiento fraccionario se pueden buscar dentro de una ventana) para lograr una búsqueda más eficaz pero de baja complejidad. En este ejemplo, el intervalo de una ventana de búsqueda (por ejemplo, con un tamaño de 16x16) puede estar predefinido o señalado en el flujo de bits.

El algoritmo de búsqueda para la búsqueda puede estar predefinido, tal como una búsqueda completa, una búsqueda de tres etapas, una búsqueda en rombo, un algoritmo de búsqueda en descenso de gradiente basado en bloques (BBGDS) como se describe, por ejemplo, en Lurug-Kuo Liu, Ephraim Feig, "A block-based gradient descent search algorithm for block motion estimation in video coding", IEEE Trans. Circuits Syst. Video Technol., vol. 6, págs., 419-422, agosto de 1996, o un algoritmo de búsqueda en rombo con sesgo central sin restricciones (UCBDS) como se describe, por ejemplo, en Jo Yew Tham, Surendra Ranganath, Maitreya Ranganath y Ashraf Ali Kassim, "A novel unrestricted center-biased diamond search algorithm for block motion estimation", IEEE Trans. Circuits Syst. Video Technol., vol. 8, págs. 369-377, agosto de 1998.

En algunos casos, se pueden usar diferentes técnicas de búsqueda en diferentes casos en base a criterios de señalización o predefinidos. Por ejemplo, para buscar una PU completa, se puede usar una búsqueda en rombo. Para buscar una sub-PU, se puede usar una pequeña búsqueda en rombo. Adicionalmente, o de forma alternativa, se puede aplicar una parada anticipada durante la búsqueda, por ejemplo, cuando el coste de coincidencia está por debajo de un umbral predefinido o adaptativo. Cuando se usa la coincidencia de plantillas, se puede aplicar además una restricción de que los dos vectores de movimiento de las dos listas de referencia después del refinamiento no apunten al mismo bloque en la misma imagen de referencia.

Como se indica anteriormente, la presente divulgación describe determinadas técnicas de optimización para los procesos de DMVD existentes. Por ejemplo, como se describe anteriormente con respecto al ejemplo de la FIG. 10, las técnicas incluyen extender la coincidencia bilateral de la predicción bidireccional a la predicción unidireccional. También se puede aplicar la coincidencia bilateral cuando las posiciones temporales de dos imágenes de referencia están ambas antes o después de la posición temporal del fragmento actual (es decir, los valores de POC son más pequeños o más grandes que los de la imagen actual). Dichas técnicas se pueden denominar conjuntamente en el presente documento coincidencia bilateral extendida.

Otros aspectos de la presente divulgación se relacionan con las técnicas de interpolación. Por ejemplo, de acuerdo con los aspectos de la presente divulgación, las técnicas de interpolación se pueden simplificar para reducir la complejidad. Como se indica anteriormente con respecto a la FIG. 2, la búsqueda de movimiento se realiza típicamente usando precisión de subpíxeles. En consecuencia, se necesita interpolación para posiciones de píxeles no enteros. Para reducir la complejidad informática, de acuerdo con aspectos de la presente divulgación, un codificador de vídeo puede usar un filtro de interpolación con tomas más cortas en comparación con la interpolación de compensación de movimiento habitual. En un ejemplo, el codificador de vídeo puede usar un filtro de interpolación bilineal durante la búsqueda de movimiento, por ejemplo, cuando se aplican vectores de movimiento candidatos iniciales o se refinan dichos vectores de movimiento. En otro ejemplo, el codificador de vídeo se puede configurar para realizar inicialmente una búsqueda de movimiento usando posiciones de píxeles

enteros, seguido de realizar una búsqueda de movimiento en posiciones de píxeles fraccionarios con la mejor coincidencia de la posición de píxeles enteros como punto de partida.

5 Otras técnicas de la presente divulgación se relacionan con la derivación de información de movimiento para subbloques. Por ejemplo, de acuerdo con los aspectos de la presente divulgación, un codificador de vídeo puede dividir un bloque/PU en sub-PU/subbloques (no superpuestos) en base a cualquier técnica de derivación de movimiento. Cuando una PU se divide en múltiples sub-PU de tamaño más pequeño, el codificador de vídeo puede derivar un conjunto único de información de movimiento para cada sub-PU.

10 En un ejemplo con propósitos ilustrativos, una PU de 32x32 se puede dividir en 16 sub-PU de 8x8. En este ejemplo, el codificador de vídeo puede determinar diferentes índices de referencia y/o vectores de movimiento para cada una de las sub-PU de 8x8. En otros ejemplos, las sub-PU pueden tener otros tamaños, por ejemplo, 4x4, 2x2 o 1x1.

15 En algunos casos, el tamaño del subbloque/sub-PU puede estar predefinido y fijo independientemente del tamaño del bloque/PU. En otros ejemplos, se puede definir una profundidad de división D para las PU que controla el número de veces que una PU se puede dividir de acuerdo con una estructura de árbol cuaternario. En algunos ejemplos, un tamaño mínimo de sub-PU/subbloque puede estar predefinido o señalizarse para indicar el tamaño objetivo del subbloque/sub-PU en el que se dividirá el bloque/PU actual. El tamaño objetivo puede ser el más grande entre el tamaño mínimo de sub-PU/subbloque y el tamaño obtenido dividiendo el bloque actual D veces de acuerdo con una estructura de árbol cuaternario.

25 De acuerdo con los aspectos de la presente divulgación, un codificador de vídeo puede aprovechar la información de movimiento derivada para una PU cuando se deriva información de movimiento para las sub-PU de la PU. Por ejemplo, el codificador de vídeo puede, para cada sub-PU, buscar información de movimiento única respectiva estableciendo la información de movimiento del bloque completo como el centro de búsqueda (punto de búsqueda inicial). A continuación, el codificador de vídeo puede refinar el movimiento para cada sub-PU. De forma alternativa, el centro de búsqueda de cada sub-PU se puede derivar de una lista de candidatos de punto de partida.

30 En otro ejemplo, se puede generar una lista de candidatos de vector de movimiento para la sub-PU usando cualquiera de las técnicas descritas en el presente documento. A continuación, el codificador de vídeo puede verificar cada candidato en la lista después de verificar el centro de búsqueda (por ejemplo, el punto de búsqueda inicial derivado de la PU). De acuerdo con aspectos de la presente divulgación, las sub-PU se pueden refinar usando cualquiera de las técnicas descritas en el presente documento que son aplicables a bloques normales. En otros ejemplos, el refinamiento se puede omitir siempre o condicionalmente para las sub-PU después de verificar la lista de candidatos de vector de movimiento para la mejor coincidencia de la manera descrita anteriormente. Una condición posible puede ser que la mejor coincidencia sigue siendo el candidato indicado por la información de movimiento de la PU (por ejemplo, el punto de búsqueda inicial) después de verificar la lista de candidatos. Si es verdadero, el codificador de vídeo puede omitir el refinamiento. De otro modo, el codificador de vídeo puede realizar el refinamiento.

45 En algunos ejemplos, el filtrado de vectores de movimiento se puede realizar para sub-PU para corregir un vector de movimiento incorrecto aislado. Por ejemplo, un codificador de vídeo puede usar un filtro mediano con el vector de movimiento de la sub-PU actual y los vectores de movimiento de hasta cuatro sub-PU vecinas. De acuerdo con aspectos de la presente divulgación, cuando se aplican transformaciones, el codificador de vídeo puede considerar la PU completa como un bloque completo, de modo que una transformada puede cruzar los límites de la sub-PU. En otros ejemplos, el codificador de vídeo puede aplicar transformadas a cada sub-PU de modo que el tamaño de transformada no sea más grande que el tamaño de la sub-PU. En algunos casos, cuando se usa la coincidencia de plantillas, también se puede dividir una PU completa en sub-PU más pequeñas. Para las sub-PU con vecinos espaciales que están todos en la PU actual (sus plantillas no están disponibles), el codificador de vídeo puede establecer sus vectores de movimiento en los vectores de movimiento derivados de la PU completa.

55 En algunos ejemplos, un codificador de vídeo puede codificar elementos de sintaxis separados que indican si un bloque particular está dividido. En otro ejemplo, todos los bloques de 2Nx2N con modo de coincidencia bilateral extendida se dividen además en divisiones pequeñas y no se señala ningún indicador adicional. El tamaño del subbloque puede estar predefinido o señalado. De forma alternativa, el tamaño del subbloque se puede basar en el tamaño del bloque actual. Por ejemplo, el tamaño del subbloque se puede derivar como el número más grande de dos valores. El primer valor es un tamaño de subbloque mínimo predefinido o señalado (por ejemplo, tal como de 4x4 u 8x8). El segundo valor es un tamaño relativo con respecto al bloque actual, tal como  $(S \gg d) \times (S \gg d)$  donde SxS es el tamaño de bloque actual, mientras que d es un valor predefinido o señalado para indicar la profundidad de árbol cuaternario con la que el bloque actual se divide en subbloques. En los ejemplos anteriores, se debe entender que el término PU se usa de manera intercambiable con el término bloque y el término sub-PU se usa de manera intercambiable con el término subbloque.

65 Como se indica anteriormente, las técnicas descritas en el presente documento se pueden usar independientemente o en combinación. Como ejemplo, un proceso de derivación de información de movimiento

asociado con un modo de derivación de información de movimiento (por ejemplo, un modo de coincidencia bilateral extendida, un modo de coincidencia de plantillas o cualquier otro modo) puede incluir tres etapas, aunque la tercera etapa se puede realizar condicionalmente en base al tipo de fragmento, nivel temporal, tipo de bloque, tamaño de bloque o sintaxis definida en el flujo de bits.

En una primera etapa, un codificador de vídeo puede formular una lista de candidatos inicial de conjuntos unipredictivos de información de movimiento para un bloque actual. El codificador de vídeo puede seleccionar el candidato que tiene el mejor punto de partida de la lista, por ejemplo, en base a un coste de coincidencia. En una segunda etapa, el codificador de vídeo puede refinar el candidato seleccionado para derivar la información de movimiento para el bloque actual. El refinamiento también se puede basar en un coste de coincidencia, como se describe en el presente documento. A continuación, el codificador de vídeo puede opcionalmente dividir además el bloque actual en subbloques como se describe en el presente documento, y refinar cada vector de movimiento para cada subbloque. Finalmente, el codificador de vídeo puede aplicar compensación de movimiento para el bloque actual usando la información de movimiento derivada. Si se ha realizado la división, el codificador de vídeo puede aplicar compensación de movimiento subbloque por subbloque.

En un ejemplo, se puede usar el siguiente pseudocódigo para reconstruir un bloque usando un modo de derivación de información de movimiento, donde la información de movimiento se deriva por coincidencia bilateral o bien coincidencia de plantillas. Cuando se usa la coincidencia bilateral, el bloque se divide además en subbloques (téngase en cuenta que MV aquí incluye información de referencia):

Reconstrucción del bloque B (con tamaño W×H) en FRUC

**Construir lista inicial**

**si B es coincidencia bilateral**

{

**Encontrar la mejor coincidencia (por ejemplo, coincidencia bilateral) en la lista inicial como punto de partida con la medición de la coincidencia bilateral**

**Refinar el MV en base al punto de partida con coincidencia bilateral para obtener el vector de movimiento  $MV_B$  para el bloque B**

**para cada subbloque en el bloque B**

{

**tomando  $MV_B$  como punto de partida, refinar MV para cada subbloque**

**realizar compensación de movimiento para el subbloque con la información de MV derivada**

}

}

**si no // coincidencia de plantillas**

{

**Encontrar la mejor coincidencia (por ejemplo, coincidencia de plantillas) en la lista inicial como punto de partida con la medición de coincidencia de plantillas**

**Refinar el MV en base al punto de partida con la coincidencia de plantillas**

}

De ahí, de acuerdo con los aspectos de la presente divulgación, el codificador de vídeo 20 o el decodificador de vídeo 30 puede determinar un modo de interpolación de información de movimiento para determinar la información de movimiento para un bloque actual (un bloque que se está codificando o decodificando actualmente). Usando el modo de interpolación de información de movimiento (por ejemplo, coincidencia bilateral, coincidencia de plantillas u otra técnica), el codificador de vídeo 20 o el decodificador de vídeo 30 puede determinar el mejor candidato de vector de movimiento en una lista de candidatos de vector de movimiento, por ejemplo, el vector de movimiento que identifica un bloque de referencia que coincide de forma estrecha con el bloque actual. El

codificador de vídeo 20 o el decodificador de vídeo 30 puede usar el candidato de vector de movimiento para identificar una ventana de búsqueda en una imagen de referencia.

El codificador de vídeo 20 o el decodificador de vídeo 30 puede refinar el candidato de vector de movimiento en base a un bloque de referencia en la ventana de búsqueda que coincide de forma estrecha con el bloque actual. Es decir, el codificador de vídeo 20 o el decodificador de vídeo 30 puede determinar un nuevo vector de movimiento interpolado para el bloque actual en base al movimiento entre el bloque de referencia en la ventana de búsqueda que coincide de forma estrecha con el bloque actual y el bloque actual. A continuación, el codificador de vídeo 20 o el decodificador de vídeo 30 puede realizar una compensación de movimiento para el bloque actual usando el vector de movimiento interpolado.

En algunos casos, el codificador de vídeo 20 o el decodificador de vídeo 30 puede dividir el bloque actual en más de un subbloque con propósitos de predicción. Además, en otros ejemplos, el codificador de vídeo 20 o el decodificador de vídeo 30 puede realizar más, menos o una disposición diferente de técnicas para interpolar información de movimiento.

De ahí que determinadas técnicas de la presente divulgación se pueden generalizar como una herramienta de codificación a nivel de bloque que aprovecha determinados conceptos de FRUC, dado el supuesto de que se puede considerar que un bloque actual de una imagen actual se predice por imágenes de referencia de una manera similar a que se puede considerar que una imagen actual se interpola por imágenes de referencia en FRUC. En un ejemplo, solo se usan los procesos basados en movimiento para la herramienta de codificación a nivel de bloque. En otro ejemplo, solo se usan los procesos basados en píxeles para la herramienta de codificación a nivel de bloque. En otro ejemplo, se usan los procesos basados en movimiento o bien los procesos basados en píxeles para un bloque dado. En otro ejemplo, tanto los procesos basados en píxeles como los procesos basados en movimiento se usan para la herramienta de codificación a nivel de bloque. En otro ejemplo, se puede reutilizar o predecir otra sintaxis a partir de las otras tramas temporales y se puede usar para la herramienta de codificación, tal como información del árbol de codificación, información de SAO, ALF, RQT.

La FIG. 12 es un diagrama de flujo que ilustra un proceso de ejemplo para determinar un modo de derivación de información de movimiento para codificar un bloque de datos de vídeo. El proceso de ejemplo de la FIG. 12 se describe con respecto a un codificador de vídeo, que se puede configurar como codificador de vídeo 20, decodificador de vídeo 30 u otro procesador.

En el ejemplo de la FIG. 12, el codificador de vídeo puede seleccionar un modo de derivación de movimiento de una pluralidad de modos de derivación de movimiento (200). En general, cada uno de los modos de derivación de movimiento puede incluir realizar una búsqueda de movimiento para un primer conjunto de datos de referencia que corresponde a un segundo conjunto de datos de referencia fuera del bloque actual. Por ejemplo, con respecto a la coincidencia de plantillas, el codificador de vídeo puede realizar una búsqueda de movimiento para identificar una plantilla en una imagen actual (por ejemplo, un primer conjunto de datos de referencia) que corresponde a la plantilla en una imagen de referencia (por ejemplo, un segundo conjunto de datos de referencia). En otro ejemplo, con respecto al movimiento bilateral, el codificador de vídeo puede realizar una búsqueda de movimiento para identificar un bloque de referencia en una primera imagen de referencia (por ejemplo, un primer conjunto de datos de referencia) que corresponde a un segundo bloque de referencia en una segunda imagen de referencia (por ejemplo, un segundo conjunto de datos de referencia). Los modos de derivación de movimiento de ejemplo pueden incluir un modo de estimación de movimiento unilateral, un modo de coincidencia bilateral, un modo de coincidencia de plantillas o un modo con base especular.

De acuerdo con algunos aspectos, el codificador de vídeo puede seleccionar el modo de derivación de información de movimiento de acuerdo con uno o más elementos de sintaxis incluidos en un flujo de bits. Por ejemplo, un decodificador de vídeo puede analizar y decodificar el uno o más elementos de sintaxis del flujo de bits y determinar el modo de derivación de información de movimiento en base a la sintaxis. Un codificador de vídeo puede someter a prueba la pluralidad de modos de derivación de información de movimiento, seleccionar el modo que tiene el mejor coste de RD y codificar el uno o más elementos de sintaxis en el flujo de bits que indican el modo seleccionado.

El codificador de vídeo puede determinar información de movimiento para el bloque usando el modo de derivación de movimiento seleccionado, que puede incluir determinar una información de movimiento inicial usando el modo seleccionado (202) y usando un proceso de optimización para refinar la información de movimiento inicial (204). Por ejemplo, el codificador de vídeo puede construir una lista de candidatos de vector de movimiento con candidatos para determinar la información de movimiento inicial. La información de movimiento inicial puede proporcionar un punto de partida para refinar la información de movimiento, como se describe en el presente documento.

A continuación, el codificador de vídeo puede codificar el bloque usando la información de movimiento determinada y sin codificar la sintaxis representativa de la información de movimiento (206). Por ejemplo, en los casos en los que el codificador de vídeo comprende un decodificador de vídeo, el decodificador de vídeo puede determinar

un bloque de referencia en una imagen de referencia en base a la información de movimiento determinada, descodificar los datos residuales de un flujo de bits codificado y combinar el residuo descodificado y el bloque de referencia determinado para reconstruir el bloque actual. En los casos en los que el codificador de vídeo comprende un codificador de vídeo, el codificador de vídeo puede codificar los datos residuales para el bloque actual en un flujo de bits codificado sin codificar los índices de referencia o vectores de movimiento para los datos residuales.

La FIG. 13 es un diagrama de flujo que ilustra un proceso de ejemplo para derivar un vector de movimiento para codificar un bloque de datos de vídeo. El proceso de ejemplo de la FIG. 13 se describe con respecto a un codificador de vídeo, que se puede configurar como codificador de vídeo 20, descodificador de vídeo 30 u otro procesador.

El codificador de vídeo puede generar una lista de vectores de movimiento candidatos (210). La lista de vectores de movimiento candidatos puede incluir uno o más candidatos de vector de movimiento que se pueden usar para derivar la información de movimiento para el bloque actual. En algunos ejemplos, los candidatos de vector de movimiento se pueden determinar a partir de bloques vecinos espacialmente, bloques temporales o de otras ubicaciones.

El codificador de vídeo puede determinar un candidato de la lista para derivar la información de movimiento (212). En algunos ejemplos, el codificador de vídeo puede realizar uno o más cálculos de coste para determinar el candidato usando un modo de derivación de movimiento particular. Por ejemplo, el codificador de vídeo puede determinar un coste de coincidencia para un primer conjunto de datos de referencia y un segundo conjunto de datos de referencia, que puede incluir un coste asociado con los vectores de movimiento respectivos, como se describe en el presente documento.

A continuación, el codificador de vídeo puede determinar el vector de movimiento derivado en base al candidato determinado (214). Por ejemplo, el codificador de vídeo puede refinar el candidato determinado para determinar el vector de movimiento derivado usando el modo de derivación de movimiento. A continuación, el codificador de vídeo puede codificar el bloque usando el vector de movimiento derivado (216). Por ejemplo, en los casos en los que el codificador de vídeo comprende un descodificador de vídeo, el descodificador de vídeo puede determinar un bloque de referencia en una imagen de referencia en base al vector de movimiento derivado, descodificar los datos residuales de un flujo de bits codificado y combinar el residuo descodificado y el bloque de referencia determinado para reconstruir el bloque actual. En los casos en los que el codificador de vídeo comprende un codificador de vídeo, el codificador de vídeo puede codificar los datos residuales para el bloque actual en un flujo de bits codificado sin codificar el vector de movimiento derivado.

La FIG. 14 es un diagrama de flujo que ilustra un proceso de ejemplo para derivar información de movimiento para subbloques de un bloque de datos de vídeo. El proceso de ejemplo de la FIG. 14 se describe con respecto a un codificador de vídeo, que se puede configurar como codificador de vídeo 20, descodificador de vídeo 30 u otro procesador.

El codificador de vídeo puede derivar información de movimiento para un bloque actual (220). En algunos ejemplos, el codificador de vídeo puede usar cualquier combinación de técnicas de derivación de información de movimiento descritas en el presente documento para derivar la información de movimiento. Por ejemplo, el codificador de vídeo puede usar cualquiera de las técnicas descritas en el presente documento para realizar una búsqueda de movimiento para un primer conjunto de datos de referencia que corresponde a un segundo conjunto de datos de referencia fuera del bloque actual (por ejemplo, coincidencia de plantillas, coincidencia bilateral o similares).

De acuerdo con aspectos de la presente divulgación, el codificador de vídeo también puede dividir el bloque en una pluralidad de subbloques (222). El codificador de vídeo puede derivar por separado la información de movimiento para subbloques respectivos que comprende realizar una búsqueda de movimiento para un primer conjunto de datos de referencia que corresponde a un segundo conjunto de datos de referencia fuera de cada subbloque respectivo (224). De nuevo, el codificador de vídeo puede usar cualquier técnica descrita en el presente documento para derivar la información de movimiento tal como, por ejemplo, coincidencia de plantillas, coincidencia bilateral o similares. Por ejemplo, el codificador de vídeo puede usar el vector de movimiento derivado como punto de partida para derivar la información de movimiento para cada uno de los subbloques, y puede refinar además la información de movimiento derivada usando cualquier combinación de las técnicas de derivación de información de movimiento descritas en el presente documento.

A continuación, el codificador de vídeo puede codificar cada uno de los subbloques en base a la información de movimiento derivada sin codificar elementos de sintaxis representativos de la información de movimiento (226). Por ejemplo, en los casos en los que el codificador de vídeo comprende un descodificador de vídeo, el descodificador de vídeo puede determinar un bloque de referencia en una imagen de referencia para cada subbloque en base a la información de movimiento determinada, descodificar los datos residuales para cada subbloque de un flujo de bits codificado y combinar el residuo descodificado y el bloque de referencia determinado para reconstruir cada subbloque. En los casos en los que el codificador de vídeo comprende un codificador de



vídeo, el codificador de vídeo puede codificar los datos residuales para cada subbloque en un flujo de bits codificado sin codificar los índices de referencia o vectores de movimiento para los datos residuales.

Se ha de reconocer que, dependiendo del ejemplo, determinadas acciones o acontecimientos de cualquiera de las técnicas descritas en el presente documento se pueden realizar en una secuencia diferente, se pueden añadir, fusionar o excluir por completo (por ejemplo, no todas las acciones o acontecimientos descritos son necesarios para la puesta en práctica de las técnicas). Además, en determinados ejemplos, las acciones o acontecimientos se pueden realizar simultáneamente, por ejemplo, a través de procesamiento de múltiples hilos, procesamiento por interrupciones o múltiples procesadores, en lugar de secuencialmente.

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 se pueden almacenar en, o transmitirse sobre, un medio legible por ordenador como una o más instrucciones o código, y ejecutarse por una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador que corresponden a un medio tangible, tales como medios de almacenamiento de datos, o 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 cualesquiera medios disponibles a los que se puede acceder por uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en 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 por 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 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 deberá entender que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que, en cambio, están dirigidos a medios de almacenamiento tangibles no transitorios. Los discos, como se usa 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 normalmente datos de forma magnética, mientras que otros discos reproducen datos de forma óptica con láseres. Las combinaciones de los anteriores también se deben incluir dentro del alcance de los medios legibles por ordenador.

Las instrucciones se pueden ejecutar por uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), matrices de puertas programables por campo (FPGA) u otros circuitos lógicos integrados o discretos equivalentes. 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 decodificar, o incorporar en un códec combinado. Además, las técnicas se podrían implementar por completo en uno o más circuitos o elementos lógicos.

Las técnicas de la presente divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluyendo un aparato telefónico inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En la presente 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 por diferentes unidades de hardware. En cambio, como se describe anteriormente, diversas unidades se pueden combinar en una unidad de hardware de códec o proporcionar por un grupo de unidades de hardware interoperativas, que incluyen uno o más procesadores como se describe 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 derivación de vector de movimiento de lado de descodificador para procesar datos de vídeo, comprendiendo el procedimiento:

determinar una lista de candidatos de vector de movimiento que comprende una pluralidad de vectores de movimiento candidatos para derivar información de movimiento de un bloque actual en una imagen actual de datos de vídeo, en el que la información de movimiento indica un movimiento del bloque actual en relación con los datos de vídeo de referencia, y en el que la lista de candidatos de vector de movimiento comprende uno o más vectores de movimiento candidatos en modo de fusión o uno o más vectores de movimiento candidatos en modo de predicción de vector de movimiento avanzada;

seleccionar un vector de movimiento candidato de la lista de candidatos de vector de movimiento; y  
determinar un vector de movimiento derivado para el bloque actual que apunta a un conjunto de datos de referencia de una imagen de referencia, en el que determinar el vector de movimiento derivado comprende:

realizar una búsqueda de movimiento para un primer conjunto de datos de referencia de una primera imagen de referencia correspondiente a un segundo conjunto de datos de referencia fuera del bloque actual, apuntándose a la primera imagen de referencia por el vector de movimiento candidato, en el que el vector de movimiento candidato indica un punto de partida en la primera imagen de referencia para la búsqueda de movimiento; y

refinar el vector de movimiento candidato para determinar un vector de movimiento derivado que apunta al primer conjunto de datos de referencia, en el que el primer conjunto de datos de referencia y el segundo conjunto de datos de referencia representan los datos de referencia de la mejor coincidencia, en el que la mejor coincidencia se basa en un coste de coincidencia entre el primer conjunto de datos de referencia de la primera imagen de referencia y el segundo conjunto de datos de referencia;

en el que el coste de coincidencia comprende una magnitud de una diferencia entre el vector de movimiento derivado y el vector de movimiento candidato.

2. El procedimiento de la reivindicación 1, que comprende además generar la lista de candidatos de vector de movimiento, en el que cada entrada de la lista de candidatos de vector de movimiento comprende un vector de movimiento y su índice de referencia asociado.

3. El procedimiento de la reivindicación 2, en el que generar la lista de candidatos de vector de movimiento comprende identificar uno o más vectores de movimiento a partir de un vecino espacial del bloque actual, un vecino temporal del bloque actual o derivar un vector de movimiento usando una técnica de derivación de vector de movimiento.

4. El procedimiento de la reivindicación 2, en el que generar la lista de candidatos de vector de movimiento comprende generar la lista de candidatos de vector de movimiento y una lista de candidatos de vector de movimiento en modo de fusión de la misma manera.

5. El procedimiento de la reivindicación 2, en el que generar la lista de candidatos de vector de movimiento comprende extrapolar o interpolar un campo de movimiento de la imagen que contiene el bloque actual y determinar uno o más vectores de movimiento candidatos a partir del campo de movimiento extrapolado o interpolado.

6. El procedimiento de la reivindicación 2, en el que generar la lista de candidatos de vector de movimiento comprende recortar vectores de movimiento candidatos duplicados de la lista de candidatos de vector de movimiento.

7. El procedimiento de la reivindicación 2, que comprende además ajustar a escala los vectores de movimiento candidatos usando una imagen de referencia fija de una lista de imágenes de referencia para el bloque actual, en el que la imagen de referencia fija es una imagen inicial en la lista de imágenes de referencia para el bloque actual.

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

obtener un par de vectores de movimiento para cada vector de movimiento candidato de la lista de candidatos de vector de movimiento;

realizar un proceso de coincidencia bilateral para los pares de vectores de movimiento respectivos; y

en el que seleccionar el uno o más vectores de vector de movimiento candidatos comprende seleccionar uno o más vectores de movimiento a partir del proceso de coincidencia bilateral.

9. El procedimiento de la reivindicación 8, en el que obtener el par de vectores de movimiento para cada vector de movimiento candidato comprende determinar un primer vector de movimiento del par a partir del vector de movimiento candidato respectivo y determinar un segundo vector de movimiento del par en base a una imagen de referencia de una lista de imágenes de referencia para el bloque actual.

10. El procedimiento de la reivindicación 1, en el que determinar el vector de movimiento derivado para el bloque actual en base al refinamiento del vector de movimiento candidato seleccionado comprende:

determinar una ventana de búsqueda en una imagen de referencia en base al vector de movimiento candidato; en el que la búsqueda de movimiento comprende buscar el primer conjunto de datos de referencia que mejor coincide con el segundo conjunto de datos de referencia dentro de la ventana de búsqueda determinada; y determinar el vector de movimiento derivado para el bloque actual en base a un resultado de la búsqueda.

11. El procedimiento de la reivindicación 1, que comprende además descodificar el bloque actual de datos de vídeo usando el vector de movimiento derivado, en el que descodificar el bloque actual comprende:

determinar un bloque de referencia en una imagen de referencia en base al vector de movimiento derivado; descodificar los datos residuales de un flujo de bits codificado; y combinar el residuo descodificado y el bloque de referencia determinado para reconstruir el bloque actual.

12. El procedimiento de la reivindicación 1,

en el que seleccionar el vector de movimiento candidato comprende además seleccionar el vector de movimiento candidato en base a un coste de coincidencia asociado con el vector de movimiento candidato que incluye el coste de vector de movimiento; y

en el que determinar el vector de movimiento derivado comprende buscar, como parte de la búsqueda de movimiento, el primer conjunto de datos de referencia que mejor coincide con el segundo conjunto de datos de vídeo de referencia dentro de una ventana de búsqueda indicada por el vector de movimiento candidato y determinar el vector de movimiento derivado en base a un resultado de la búsqueda.

13. Un aparato para el procesamiento de derivación de vector de movimiento de lado de descodificador de datos de vídeo, comprendiendo el aparato:

medios para determinar una lista de candidatos de vector de movimiento que comprende una pluralidad de vectores de movimiento candidatos para derivar información de movimiento de un bloque actual de una imagen actual de datos de vídeo, en el que la información de movimiento indica un movimiento del bloque actual en relación con los datos de vídeo de referencia, y en el que la lista de candidatos de vector de movimiento comprende uno o más vectores de movimiento candidatos en modo de fusión o uno o más vectores de movimiento candidatos en modo de predicción de vector de movimiento avanzada;

medios para seleccionar el vector de movimiento candidato de la lista de candidatos de vector de movimiento; y medios para determinar un vector de movimiento derivado para el bloque actual que apunta a un conjunto de datos de referencia de una imagen de referencia, en el que determinar el vector de movimiento derivado comprende:

realizar una búsqueda de movimiento para un primer conjunto de datos de referencia de una primera imagen de referencia correspondiente a un segundo conjunto de datos de referencia fuera del bloque actual, apuntándose a la imagen de referencia por el vector de movimiento candidato, en el que el vector de movimiento candidato indica un punto de partida en la primera imagen de referencia para la búsqueda de movimiento; y

refinar el vector de movimiento candidato para determinar un vector de movimiento derivado que apunta al primer conjunto de datos de referencia, en el que el primer conjunto de datos de referencia y el segundo conjunto de datos de referencia representan los datos de referencia de la mejor coincidencia, en el que la mejor coincidencia se basa en un coste de coincidencia entre el primer conjunto de datos de referencia de la primera imagen de referencia y el segundo conjunto de datos de referencia;

en el que el coste de coincidencia comprende una magnitud de una diferencia entre el vector de movimiento derivado y el vector de movimiento candidato.

14. Un medio no transitorio legible por ordenador que tiene instrucciones almacenadas en el mismo que, cuando se ejecutan, hacen que uno o más procesadores realicen el procedimiento de una cualquiera de las reivindicaciones 1 a 12.

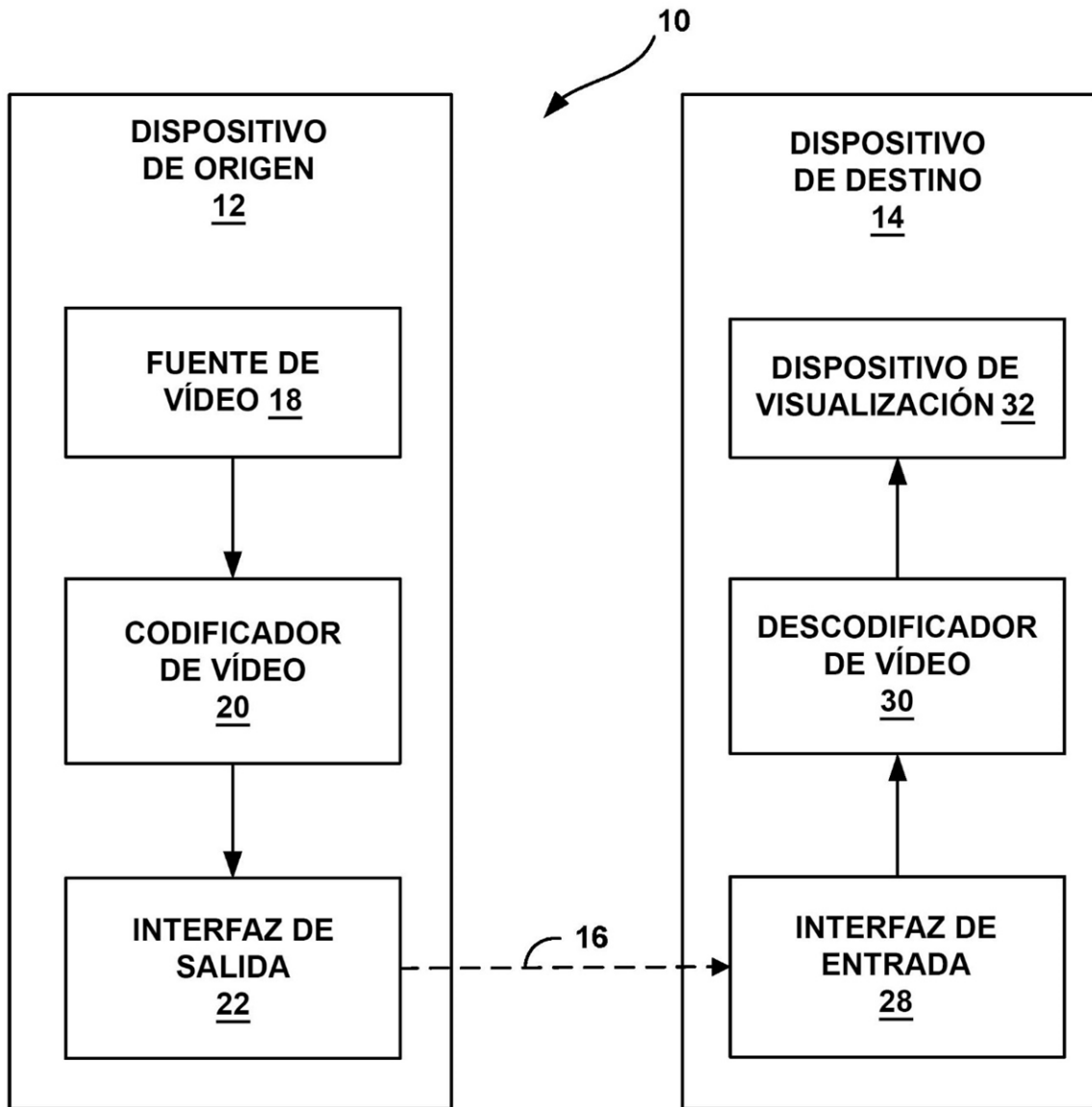


FIG. 1

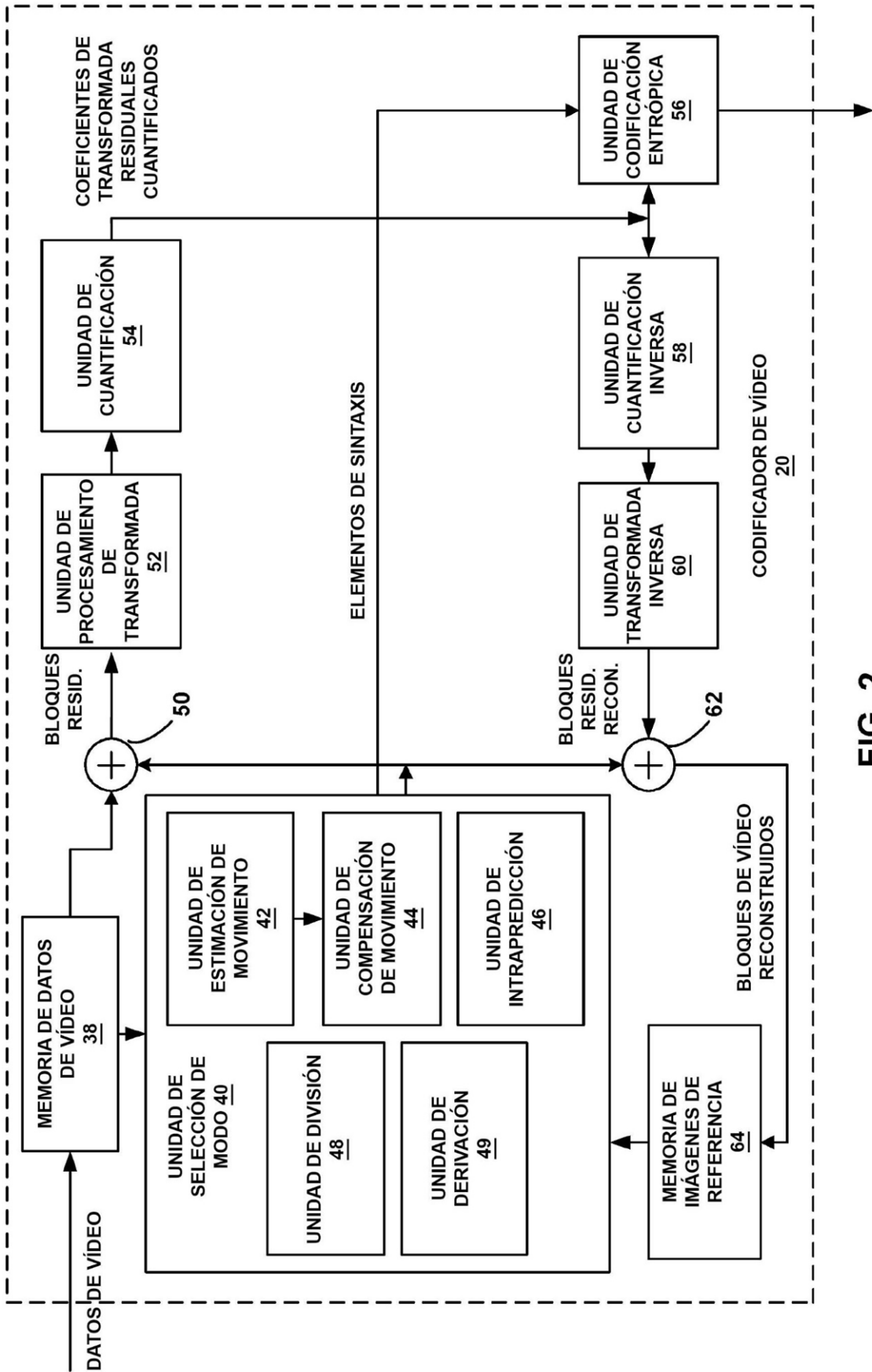


FIG. 2

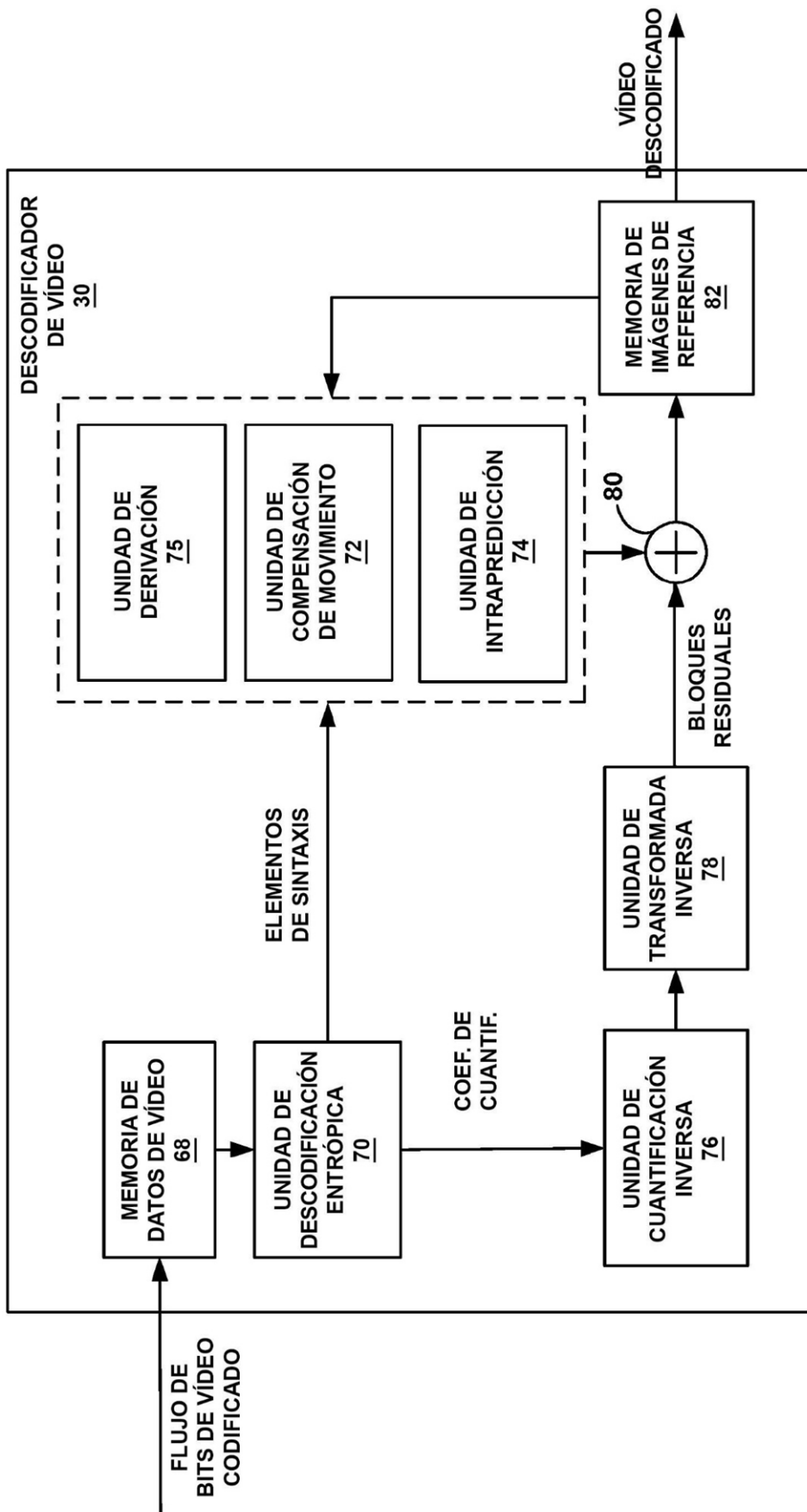
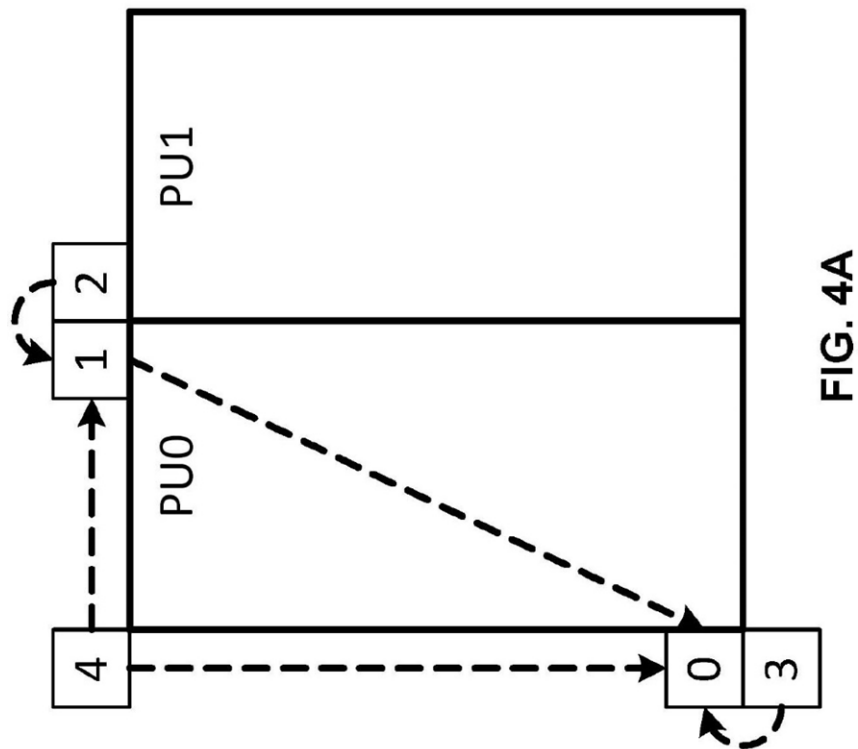
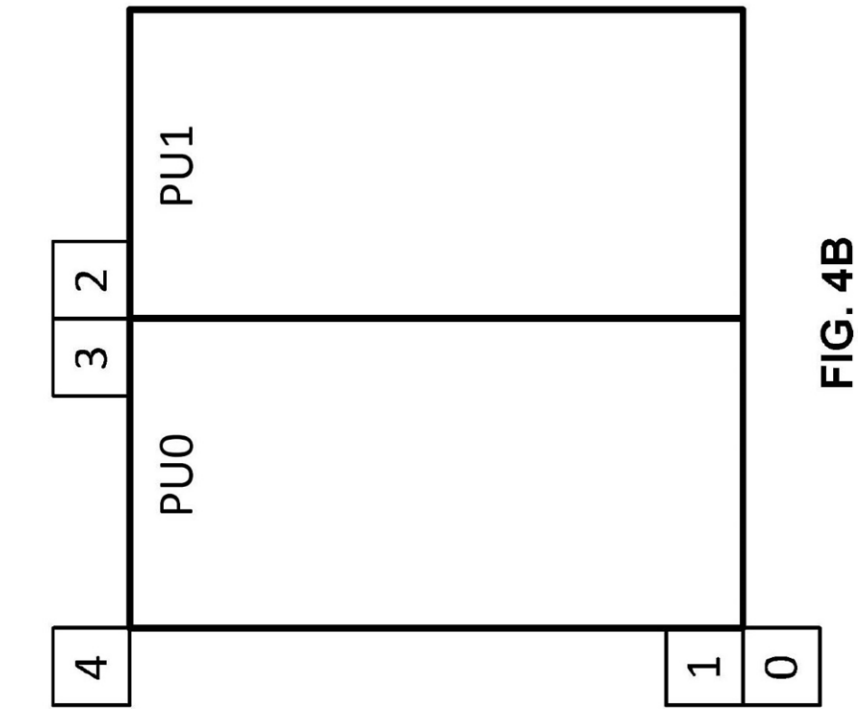


FIG. 3



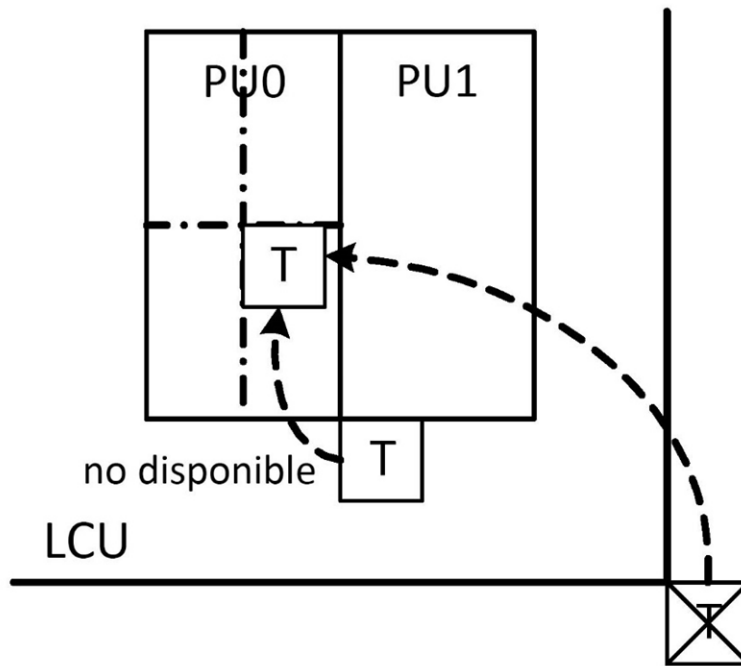


FIG. 5A

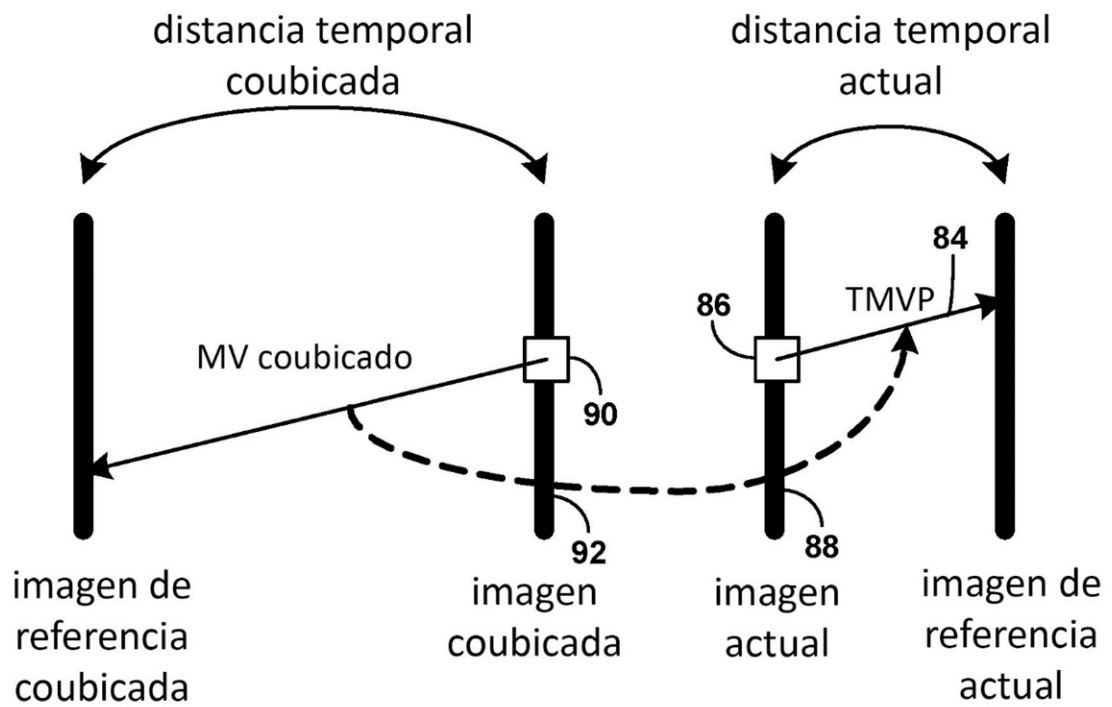
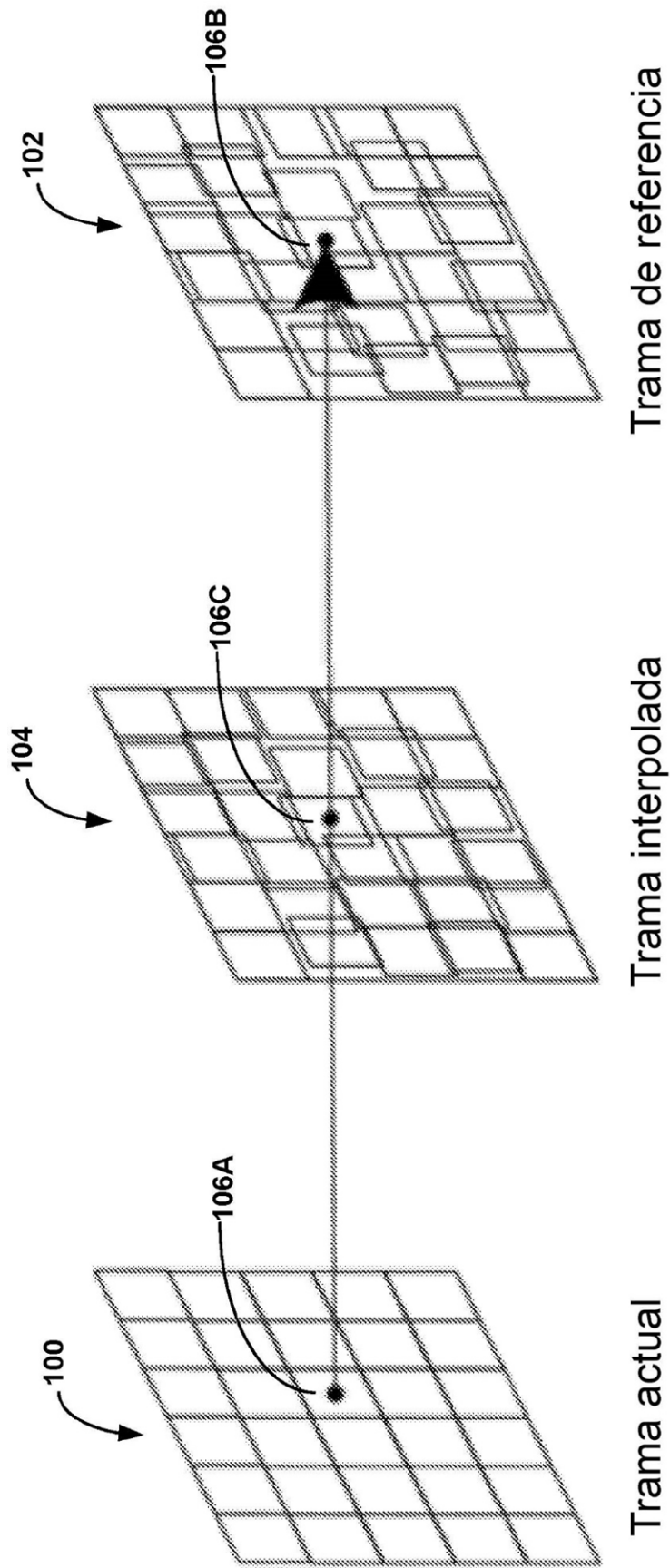


FIG. 5B





**FIG. 6**

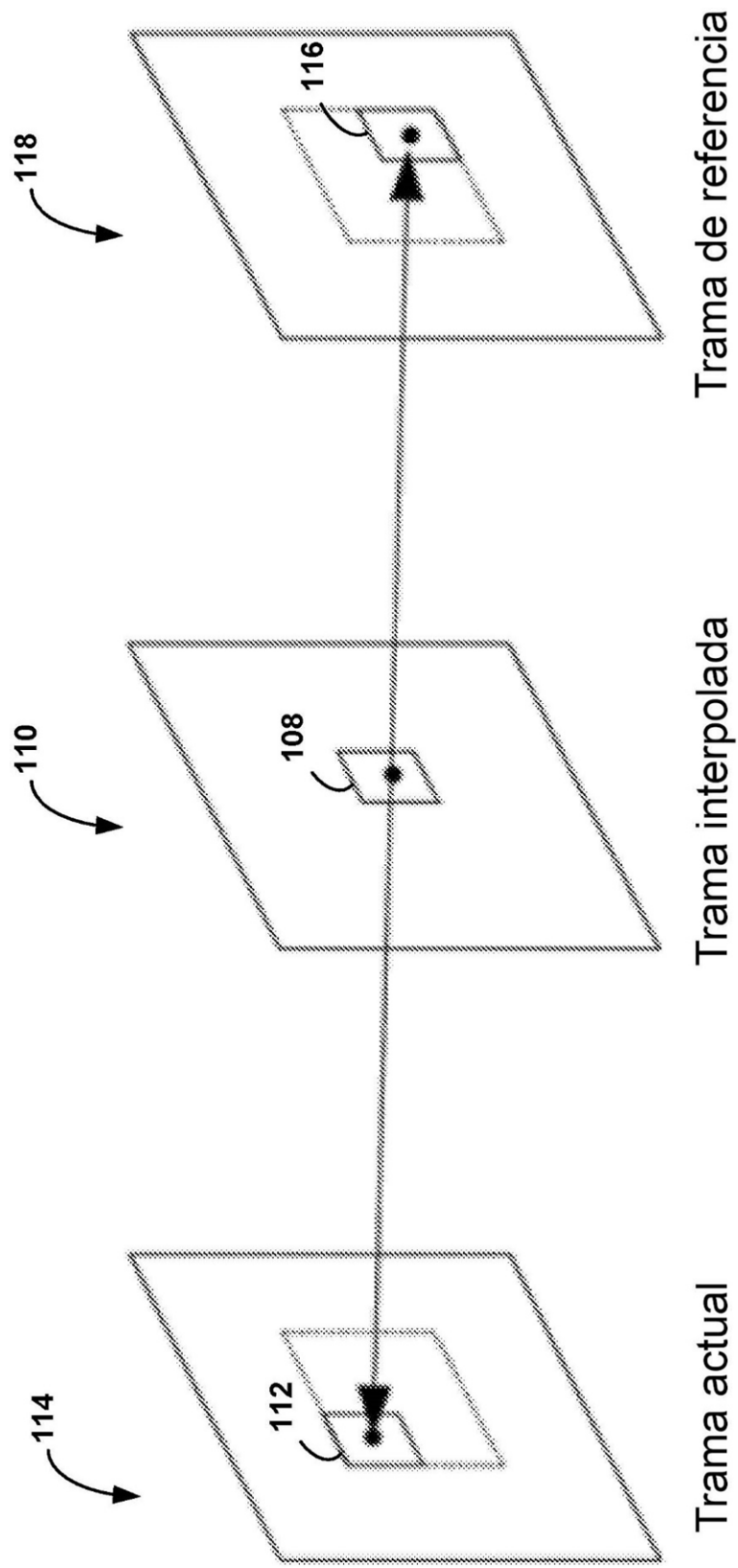


FIG. 7

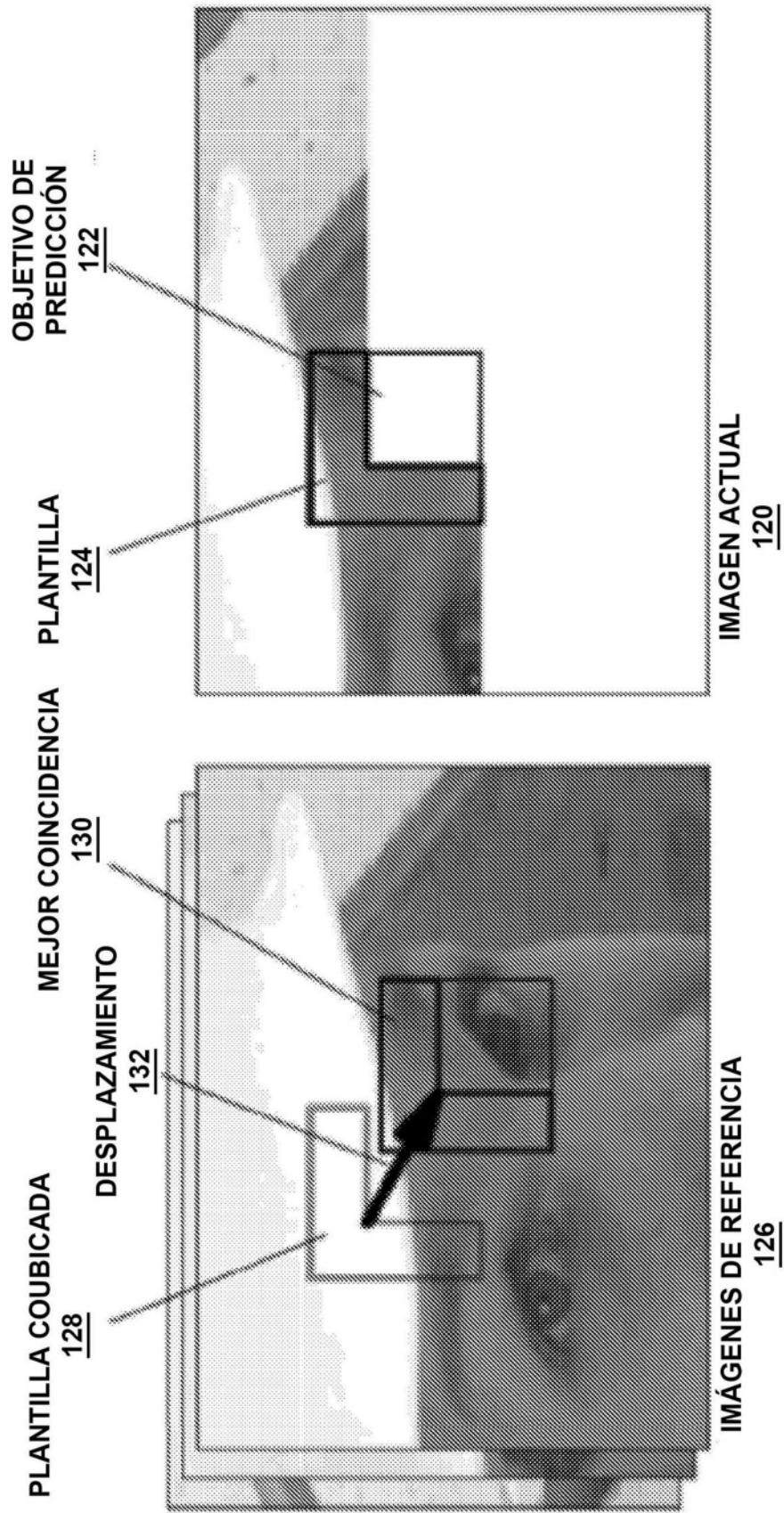


FIG. 8

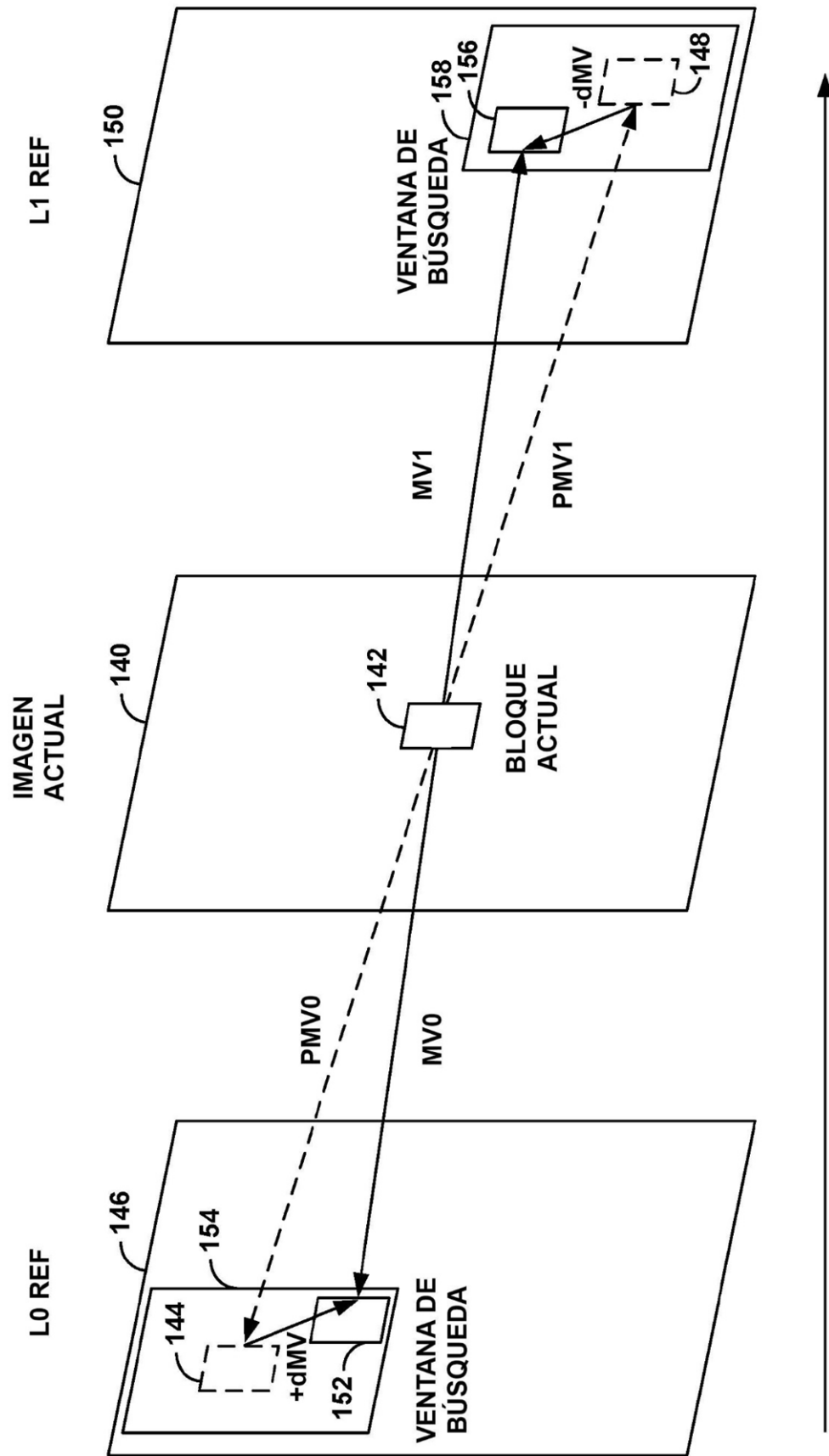


FIG. 9

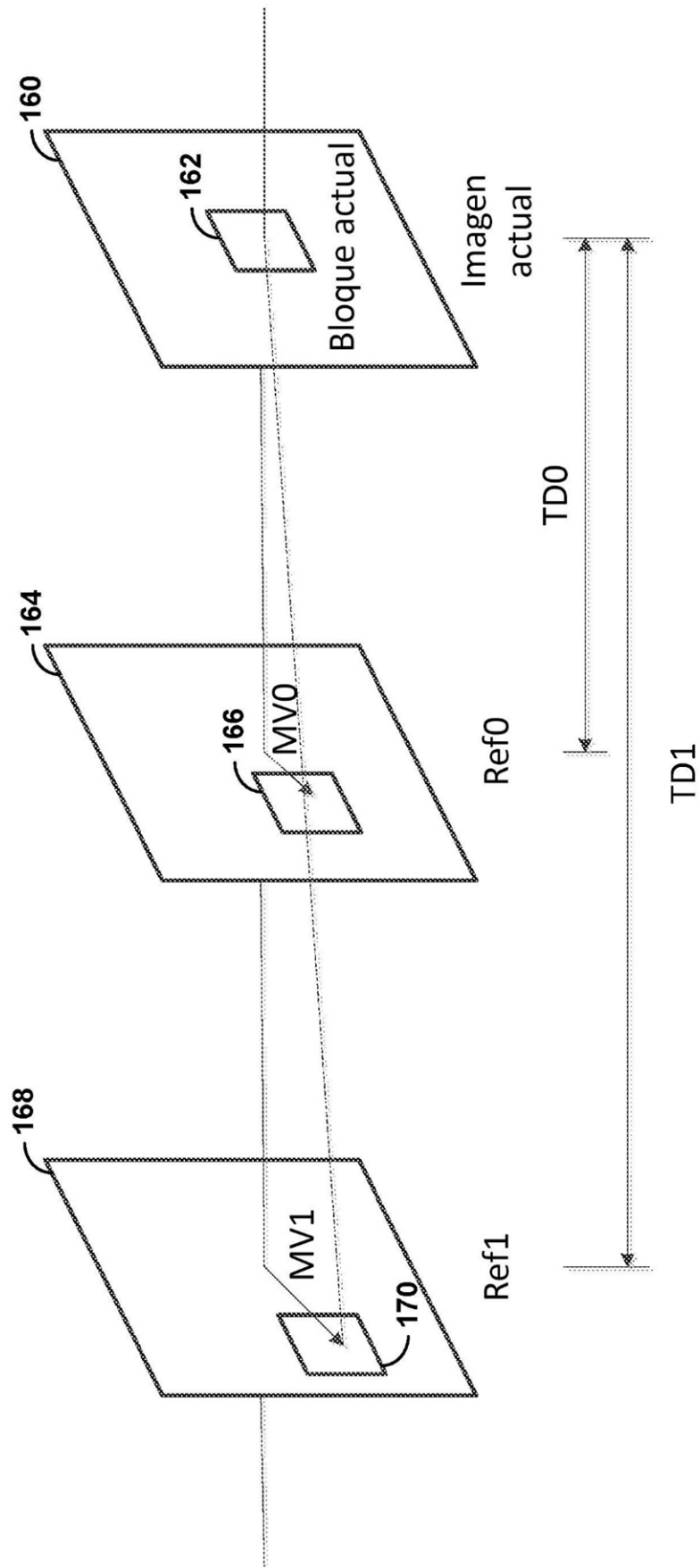


FIG. 10

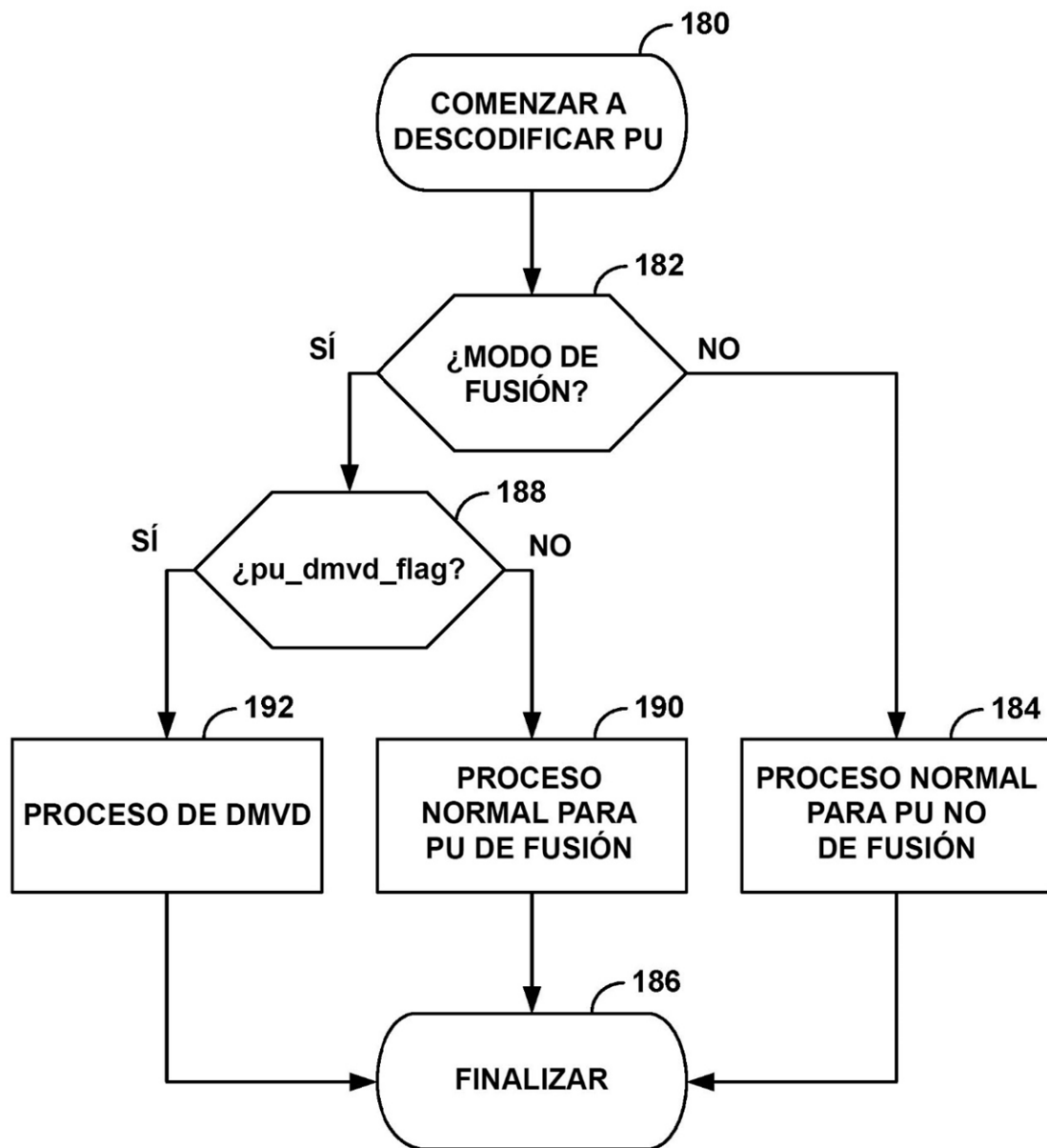
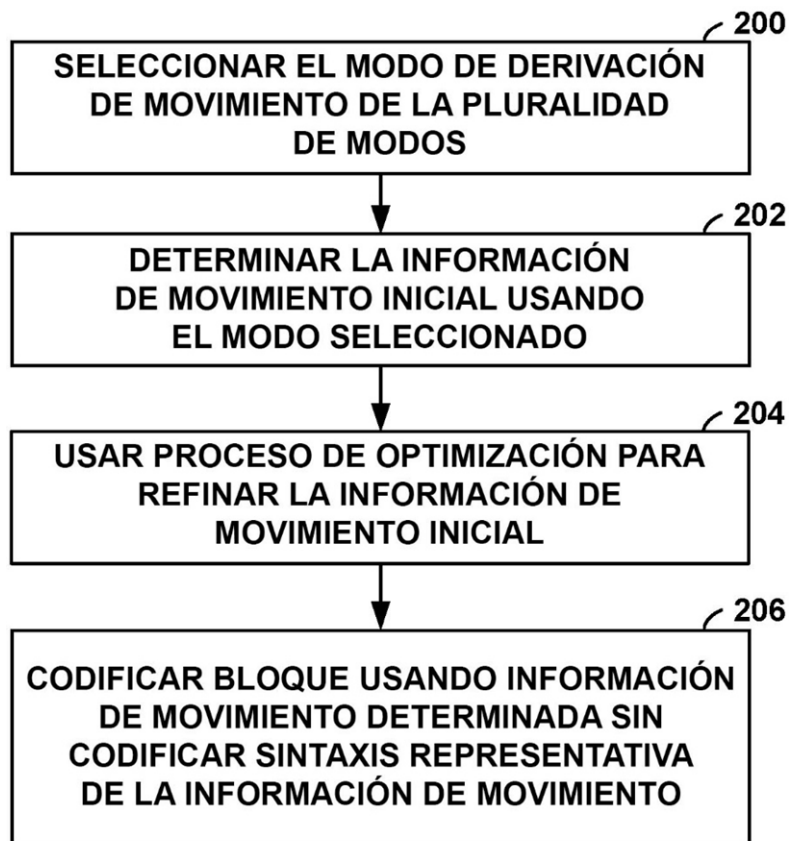
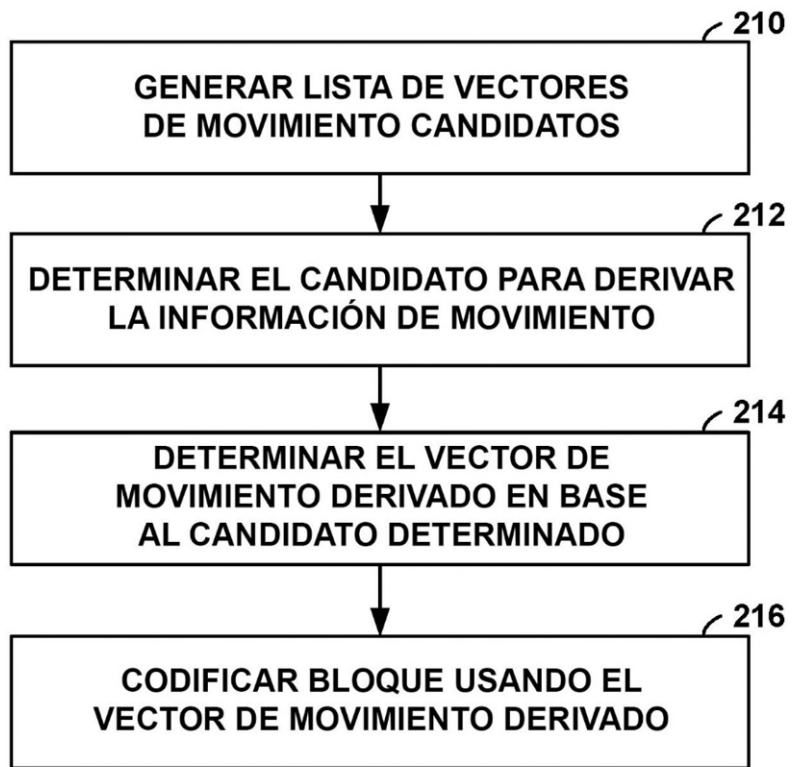


FIG. 11

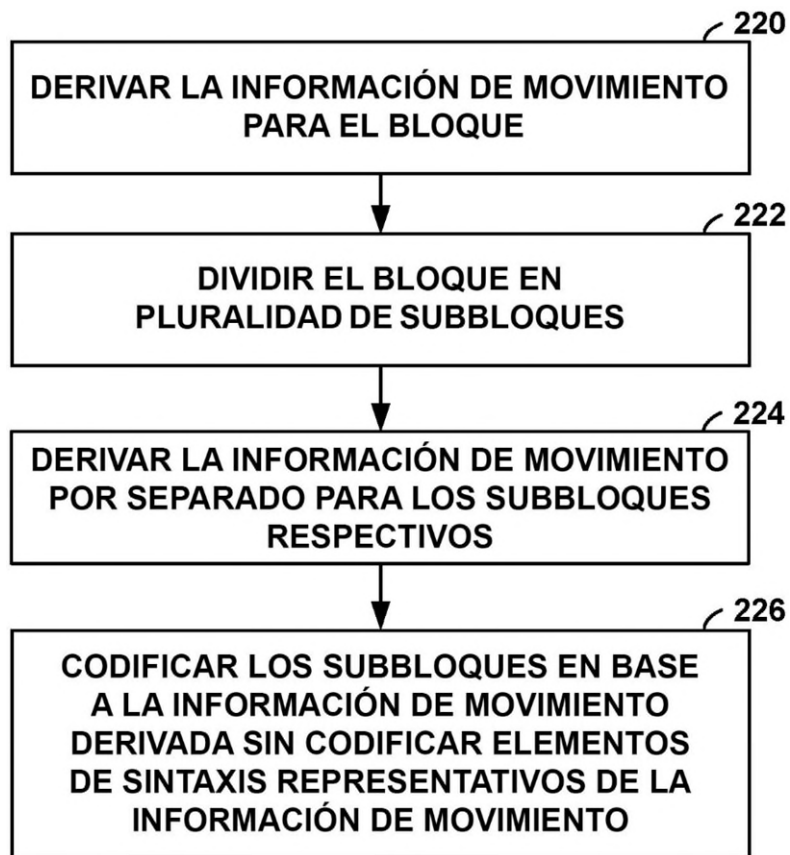


**FIG. 12**



**FIG. 13**





**FIG. 14**