

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 900 751**

51 Int. Cl.:

**H04N 19/597** (2014.01)

**H04N 19/52** (2014.01)

**H04N 19/70** (2014.01)

**H04N 19/58** (2014.01)

12

## TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **14.03.2013 PCT/US2013/031573**

87 Fecha y número de publicación internacional: **19.09.2013 WO13138639**

96 Fecha de presentación y número de la solicitud europea: **14.03.2013 E 13713300 (5)**

97 Fecha y número de publicación de la concesión europea: **24.11.2021 EP 2826248**

54 Título: **Extensiones de sintaxis de alto nivel para codificación de vídeo de alta eficacia**

30 Prioridad:

**16.03.2012 US 201261611959 P**

**16.04.2012 US 201261624990 P**

**11.06.2012 US 201261658344 P**

**22.06.2012 US 201261663484 P**

**27.12.2012 US 201261746476 P**

**13.03.2013 US 201313801731**

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

**18.03.2022**

73 Titular/es:

**QUALCOMM INCORPORATED (100.0%)**

**Attn: International IP Administration 5775**

**Morehouse Drive**

**San Diego, California 92121-1714, US**

72 Inventor/es:

**CHEN, YING;**

**WANG, YE-KUI y**

**ZHANG, LI**

74 Agente/Representante:

**ISERN JARA, Jorge**

ES 2 900 751 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

Extensiones de sintaxis de alto nivel para codificación de vídeo de alta eficacia

## 5 CAMPO TÉCNICO

Esta divulgación se refiere a la codificación de vídeo.

10 ANTECEDENTES

Las capacidades de vídeo digital se pueden incorporar a una amplia gama de dispositivos, que incluye televisores digitales, sistemas de radiodifusión directa digital, sistemas de radiodifusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, ordenadores de tipo tableta, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos de radio celulares o por satélite, los denominados "teléfonos inteligentes", dispositivos de teleconferencia por vídeo, 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) actualmente en desarrollo y ampliaciones 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.

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

La predicción espacial o temporal da como resultado un bloque predictivo para un bloque que se va a codificar. Los datos residuales representan las diferencias de píxel entre el bloque original a codificar y el bloque predictivo. Un bloque intercodificado se codifica de acuerdo con un vector de movimiento que apunta a un bloque de muestras de referencia que forman el bloque predictivo, y los datos residuales que indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intracodificado se codifica de acuerdo con un modo de intracodificación y con los datos residuales. Para una compresión adicional, los datos residuales se pueden transformar desde el dominio de píxel a un dominio de transformada, dando como resultado unos coeficientes de transformada residuales, que a continuación se pueden cuantificar. Los coeficientes de transformada cuantificados, inicialmente dispuestos en una matriz bidimensional, se pueden explorar para generar un vector unidimensional de coeficientes de transformada, y se puede aplicar codificación entrópica para conseguir todavía más compresión.

Se hace referencia a los siguientes artículos:

"Descriptions of 3D Video Coding Proposal (HEVC-Compatible Category) by Sony", de YOSHITOMO TAKAHASHI ET AL.: 98. CONFERENCIA DEL MPEG; del 28/11/2011 al 2/12/2011; GINEBRA; n.º m22566, 22 de noviembre de 2011; y  
"Restriction on motion vector scaling for Merge and AMVP" de I-K KIM ET AL.: 7. CONFERENCIA DEL JCT-VC; 98. CONFERENCIA DEL MPEG; del 21/11/2011 al 30/11/2011; GINEBRA;

## SUMARIO

La invención se define en las reivindicaciones, a las que ahora se hace referencia. La divulgación suficiente para la invención tal como se define en las reivindicaciones se puede encontrar en la FIG. 8 y en los pasajes correspondientes de la presente descripción. Las FIGS. 5, 6, 7 y 9 describen modos de realización alternativos que no pertenecen a la invención.

En general, esta divulgación describe varias técnicas para admitir ampliaciones de normas de codificación, tales como la inminente norma de codificación de vídeo de alta eficacia (HEVC), solamente con cambios de sintaxis de alto nivel. Por ejemplo, esta divulgación describe técnicas tanto en la especificación base de HEVC como en las ampliaciones de HEVC del códec de vídeo multivista y/o el códec de vídeo tridimensional (3D), donde la vista base es compatible con la especificación base de HEVC. En general, una "especificación base de codificación de vídeo" puede corresponder a una especificación de codificación de vídeo, tal como una especificación base de HEVC, que se usa para codificar datos de vídeo bidimensionales de una sola capa. Las ampliaciones de la especificación base de

codificación de vídeo pueden ampliar las capacidades de la especificación base de codificación de vídeo para permitir la codificación de vídeo en 3D y/o multicapa. La especificación base de HEVC representa un ejemplo de una especificación base de codificación de vídeo, mientras que las ampliaciones de MVC y SVC de la especificación base de HEVC representan ejemplos de ampliaciones de una especificación base de codificación de vídeo.

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

## BREVE DESCRIPCIÓN DE LOS DIBUJOS

La FIG. 1 es un diagrama de bloques que ilustra un ejemplo de un sistema de codificación y decodificación de vídeo que puede utilizar técnicas para codificar datos de vídeo de acuerdo con una ampliación solamente de sintaxis de alto nivel de una norma de codificación de vídeo.

La FIG. 2 es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo que puede implementar técnicas para codificar datos de vídeo de acuerdo con una ampliación solamente de sintaxis de alto nivel de una norma de codificación de vídeo.

La FIG. 3 es un diagrama de bloques que ilustra un ejemplo de un decodificador de vídeo que puede implementar técnicas para codificar datos de vídeo de acuerdo con una ampliación solamente de sintaxis de alto nivel de una norma de codificación de vídeo.

La FIG. 4 es un diagrama conceptual que ilustra un ejemplo de patrón de predicción MVC.

Las FIGS. 5-9 son diagramas conceptuales que ilustran posibles problemas que deben resolverse para lograr una ampliación de HEVC solamente de sintaxis de alto nivel, la FIG. 8 pertenece a la invención mientras que las FIGS. 5, 6, 7 y 9 no pertenecen a la invención.

La FIG. 10 es un diagrama conceptual que ilustra un ejemplo de conjunto de bloques vecinos a un bloque actual para su uso en la predicción de vectores de movimiento

La FIG. 11 es un diagrama de flujo que ilustra un ejemplo de procedimiento para codificar datos de vídeo de acuerdo con las técnicas de esta divulgación.

La FIG. 12 es un diagrama de flujo que ilustra un ejemplo de procedimiento para decodificar datos de vídeo de acuerdo con las técnicas de esta divulgación.

## DESCRIPCIÓN DETALLADA

En general, esta divulgación describe varias técnicas para admitir ampliaciones de normas de codificación, tales como la inminente norma de codificación de vídeo de alta eficacia (HEVC), solamente con cambios de sintaxis de alto nivel (HLS). Por ejemplo, esta divulgación describe técnicas tanto en la especificación base de HEVC como en las ampliaciones de HEVC de la codificación de vídeo multivista (MVC) y/o la codificación de vídeo tridimensional (3DV), donde la vista base es compatible con la especificación base de HEVC.

Esta divulgación describe determinadas técnicas para habilitar un perfil de solo sintaxis de alto nivel en una especificación de ampliación de HEVC. El término "entre vistas" en el contexto de MVC/3DV puede sustituirse por "entre capas" en el contexto de la codificación de vídeo escalable (SVC). Es decir, aunque la descripción de estas técnicas se centra principalmente en la predicción "entre vistas", se pueden aplicar las mismas ideas u otras similares a imágenes de referencia "entre capas" para una ampliación SVC de solo HLS de HEVC.

La FIG. 1 es un diagrama de bloques que ilustra un ejemplo de un sistema de codificación y decodificación de vídeo 10 que puede utilizar técnicas para codificar datos de vídeo de acuerdo con una ampliación solamente de sintaxis de alto nivel de una norma de codificación 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 un dispositivo de destino 14 va a decodificar en un momento posterior. 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 cualesquiera de entre una amplia gama de dispositivos, incluyendo ordenadores de escritorio, ordenadores de tipo *notebook* (es decir, ordenadores portátiles), ordenadores de tipo tableta, decodificadores multimedia, aparatos telefónicos tales como los denominados teléfonos "inteligentes", los denominados paneles "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de transmisión continua de vídeo o similares. En algunos casos, el dispositivo de origen 12 y el dispositivo de destino 14 pueden estar equipados para la comunicación inalámbrica.

El dispositivo de destino 14 puede recibir los datos de vídeo codificados que se van a decodificar por medio del medio legible por ordenador 16. El medio legible por ordenador 16 puede comprender cualquier tipo de medio o dispositivo que pueda transferir los datos de vídeo codificados desde el dispositivo de origen 12 hasta el dispositivo de destino 14. En un ejemplo, el medio legible por ordenador 16 puede comprender un medio de comunicación para permitir que el dispositivo de origen 12 transmita datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real. Los datos de vídeo codificados se pueden modular 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 físicas de transmisión. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área amplia o una red global como Internet. El medio de comunicación puede incluir encaminadores, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo de origen 12 hasta el dispositivo de destino 14.

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

Las técnicas de esta divulgación no están necesariamente limitadas a las aplicaciones o configuraciones inalámbricas. Las técnicas se pueden aplicar a la codificación de vídeo en apoyo de cualquiera de una variedad de aplicaciones multimedia, tales como radiodifusiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones continuas de vídeo por Internet, tales como la transmisión continua dinámica adaptativa a través de HTTP (DASH), vídeo digital codificado en un medio de almacenamiento de datos, descodificación de vídeo digital almacenado en un medio de almacenamiento de datos u otras aplicaciones. En algunos ejemplos, el sistema 10 puede estar configurado para admitir la transmisión de vídeo unidireccional o bidireccional, para admitir aplicaciones tales como la transmisión continua de vídeo, la reproducción de vídeo, la radiodifusión de vídeo y/o la 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 decodificador de vídeo 30 y un dispositivo de visualización 32. De acuerdo con esta divulgación, el codificador de vídeo 20 del dispositivo de origen 12 puede configurarse para aplicar las técnicas para codificar datos de vídeo de acuerdo con una ampliación solamente de sintaxis de alto nivel de una norma de codificación 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 se puede interconectar con un dispositivo de visualización externo, en lugar de incluir un dispositivo de visualización integrado.

El sistema ilustrado 10 de la FIG. 1 es simplemente un ejemplo. Las técnicas de codificación de datos de vídeo de acuerdo con una ampliación solamente de sintaxis de alto nivel de una norma de codificación de vídeo pueden ser realizadas por cualquier dispositivo de codificación y/o decodificación de vídeo digital. Aunque, en general, un dispositivo de codificación de vídeo realiza las técnicas de esta divulgación, un codificador/decodificador de vídeo, típicamente denominado "CÓDEC", también puede realizar las técnicas. Además, un preprocesador de vídeo también puede realizar las técnicas de esta divulgación. 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 una manera sustancialmente simétrica, de modo que cada uno de los dispositivos 12, 14 incluye componentes de codificación y decodificación de vídeo. Por consiguiente, el sistema 10 puede admitir la 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 transmisión de vídeo para recibir vídeo desde un proveedor de contenido de vídeo. Como otra alternativa, la fuente de vídeo 18 puede generar datos basados en gráficos de ordenador como vídeo de origen, o una combinación de vídeo en directo, vídeo archivado y vídeo generado por ordenador. En algunos casos, si la fuente de vídeo 18 es una cámara de vídeo, el dispositivo de origen 12 y el dispositivo de destino 14 pueden formar los denominados teléfonos con cámara o videoteléfonos. Sin embargo, como se menciona anteriormente, las técnicas descritas en esta divulgación pueden aplicarse a la codificación de vídeo en general, y se pueden aplicar a aplicaciones inalámbricas y/o alámbricas. En cada caso, el codificador de vídeo 20 puede codificar el vídeo captado, precaptado o generado por ordenador. La

interfaz de salida 22 puede proporcionar a continuación información de vídeo codificado 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 flash, un disco compacto, un disco de vídeo digital, un disco Blu-ray u otro medio legible 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 una transmisión de red. De forma similar, un dispositivo informático de una instalación de producción de medios, tal como una instalación de estampado 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 un medio legible por ordenador 16. La información del medio legible por ordenador 16 puede incluir información de sintaxis definida por el codificador de vídeo 20, usada también por el descodificador de vídeo 30, que incluye elementos de sintaxis que describen características y/o el procesamiento de bloques y otras unidades codificadas, por ejemplo, GOP. El dispositivo de visualización 32 muestra los datos de vídeo descodificados a un usuario, y puede comprender cualquiera de una variedad de dispositivos de visualización tales como un tubo de rayos catódicos (CRT), una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de visualización.

El codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con una norma de codificación de vídeo, tal como la norma de codificación de vídeo de alta eficacia (HEVC), actualmente en desarrollo, y pueden cumplir con el modelo de pruebas (HM) de HEVC. Un borrador reciente de HEVC, denominado "*HEVC Working Draft 7*" o "*WD7*" se describe en el documento JCTVC-I1003, de Bross *et al.*, "*High Efficiency Video Coding (HEVC) Text Specification Draft 7*", Equipo de colaboración conjunta en codificación de vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, novena conferencia: Ginebra, Suiza, del 27 de abril de 2012 al 7 de mayo de 2012, que, a fecha de 22 de junio de 2102, se puede descargar de [http://phenix.it-sudparis.eu/jct/doc\\_end\\_user/documents/9\\_Geneva/wg11/JCTVC-11003-v3.zip](http://phenix.it-sudparis.eu/jct/doc_end_user/documents/9_Geneva/wg11/JCTVC-11003-v3.zip). Como se indica anteriormente, esta divulgación incluye técnicas para ampliar HEVC usando sintaxis de alto nivel. Por consiguiente, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con una versión de HEVC ampliada usando sintaxis de alto nivel.

De forma alternativa, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con otras normas propietarias o industriales, tales como la norma ITU-T H.264, conocida alternativamente como MPEG-4, parte 10, codificación de vídeo avanzada (AVC), o ampliaciones de tales normas. Nuevamente, estas ampliaciones se pueden lograr usando sintaxis de alto nivel. Las técnicas de esta divulgación, sin embargo, no están limitadas a ninguna norma de codificación particular. Otros ejemplos de normas de codificación de vídeo incluyen MPEG-2 e ITU-T H.263. Aunque no se muestra en la FIG. 1, en algunos aspectos, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden estar integrados con un codificador y descodificador de audio, y pueden incluir unidades MUX-DEMUX adecuadas, u otro hardware y software, para gestionar la codificación tanto de audio como de vídeo en un flujo de datos común o flujos de datos distintos. Si corresponde, las unidades MUX-DEMUX pueden cumplir con el protocolo multiplexor de ITU H.223 u otros protocolos, tales como el protocolo de datagramas de usuario (UDP).

La norma UIT-T H.264/MPEG-4 (AVC) fue formulada por el Grupo de expertos en codificación de vídeo (VCEG) del UIT-T junto con el Grupo de expertos en imágenes en movimiento (MPEG) de ISO/IEC como el producto de una asociación colectiva conocida como el Equipo de vídeo conjunto (JVT). En algunos aspectos, las técnicas descritas en esta divulgación pueden aplicarse a dispositivos que generalmente cumplen con la norma H.264. La norma H.264 se describe en la recomendación ITU-T H.264, Codificación avanzada de vídeo para servicios audiovisuales genéricos, realizada por el Grupo de estudio de la ITU-T, con fecha de marzo de 2005, a la que se puede hacer referencia aquí como la norma H.264, o la especificación H.264, o la norma o especificación H.264/AVC. El Equipo de vídeo conjunto (JVT) continúa trabajando en las ampliaciones de H.264/MPEG-4 AVC.

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

El JCT-VC está trabajando en el desarrollo de la norma HEVC. Los esfuerzos de normalización de HEVC se basan en un modelo en evolución de un dispositivo de codificación de vídeo denominado modelo de pruebas (HM) de HEVC. El HM supone varias capacidades adicionales de los dispositivos de codificación de vídeo en relación con los dispositivos existentes, de acuerdo con, por ejemplo, ITU-T H.264/AVC. Por ejemplo, mientras que H.264 proporciona nueve modos de codificación mediante intrapredicción, el HM puede proporcionar hasta treinta y tres modos de codificación mediante intrapredicción.

En general, el modelo de trabajo del HM describe que una trama o imagen de vídeo se puede dividir en una secuencia de bloques de árbol o unidades máximas de codificación (LCU) que incluyen muestras tanto de luma como de croma. Los datos de sintaxis dentro de un flujo de bits pueden definir un tamaño para la LCU, que es la unidad máxima de codificación en lo que respecta al número de píxeles. Un segmento incluye una pluralidad de bloques de árbol consecutivos en orden de codificación. Una trama o imagen de vídeo se puede dividir en uno o más segmentos. Cada bloque de árbol se puede dividir en unidades de codificación (CU) de acuerdo con un árbol cuaternario. En general, una estructura de datos de árbol cuaternario incluye un nodo por CU, con un nodo raíz correspondiente al bloque de árbol. Si una CU se divide en cuatro sub-CU, el nodo correspondiente a la CU incluye cuatro nodos hoja, cada uno de los cuales corresponde a una de las sub-CU.

Cada nodo de la estructura de datos de árbol cuaternario puede proporcionar datos de sintaxis para la CU correspondiente. Por ejemplo, un nodo del árbol cuaternario puede incluir una bandera de división, que indica si la CU correspondiente al nodo está dividida en sub-CU. Los elementos de sintaxis para una CU se pueden definir de forma recursiva y pueden depender de si la CU se divide en sub-CU. Si una CU no se divide más, se denomina CU hoja. En esta divulgación, cuatro sub-CU de una CU hoja también se denominarán CU hoja incluso si no hay ninguna división explícita de la CU hoja original. Por ejemplo, si una CU de tamaño 16x16 no se divide más, las cuatro sub-CU de tamaño 8x8 también se denominarán CU hoja, aunque la CU de tamaño 16x16 no se haya dividido.

Una CU tiene un propósito similar a un macrobloque de la norma H.264, excepto que una CU no tiene una distinción de tamaño. Por ejemplo, un bloque de árbol se puede dividir en cuatro nodos hijo (también denominados sub-CU), y cada nodo hijo puede ser, a su vez, un nodo padre y dividirse en otros cuatro nodos hijo. Un nodo hijo final, no dividido, denominado nodo hoja del árbol cuaternario, comprende un nodo de codificación, también denominado CU hoja. Los datos de sintaxis asociados a un flujo de bits codificado pueden definir un número máximo de veces que un bloque de árbol se puede dividir, lo que se conoce como profundidad de CU máxima, y también pueden definir un tamaño mínimo de los nodos de codificación. En consecuencia, un flujo de bits también puede definir una unidad mínima de codificación (SCU). Esta divulgación usa el término "bloque" para referirse a cualquiera entre una CU, PU o TU, en el contexto de HEVC, o a estructuras de datos similares en el contexto de otras normas (por ejemplo, macrobloques y subbloques de los mismos en H.264/AVC).

Una CU incluye un nodo de codificación y unidades de predicción (PU) y unidades de transformada (TU) asociadas al nodo de codificación. Un tamaño de la CU corresponde a un tamaño del nodo de codificación y debe tener forma cuadrada. El tamaño de la CU puede variar desde 8x8 píxeles hasta el tamaño del bloque de árbol, con un máximo de 64x64 píxeles o más. Cada CU puede contener una o más PU y una o más TU. Los datos de sintaxis asociados a una CU pueden describir, por ejemplo, la división de la CU en una o más PU. Los modos de división pueden diferir dependiendo de cómo esté codificada la CU: en modo de omisión o directo, en modo de intrapredicción o en modo de interpredicción. Las PU se pueden dividir para que no tengan forma cuadrada. Los datos de sintaxis asociados a una CU también pueden describir, por ejemplo, la división de la CU en una o más TU de acuerdo con un árbol cuaternario. Una TU puede tener forma cuadrada o no cuadrada (por ejemplo, rectangular).

La norma HEVC permite transformaciones de acuerdo con las TU, que pueden ser diferentes para diferentes CU. Las TU tienen típicamente un tamaño basado en el tamaño de las PU dentro de una CU dada definida para una LCU dividida, aunque puede que no siempre sea así. Las TU son típicamente del mismo tamaño o más pequeñas que las PU. En algunos ejemplos, las muestras residuales correspondientes a una CU se pueden subdividir en unidades más pequeñas usando una estructura de árbol cuaternario conocida como "árbol cuaternario residual" (RQT). Los nodos hoja del RQT se pueden denominar unidades de transformada (TU). Los valores de diferencia de píxeles asociados a las TU se pueden transformar para generar coeficientes de transformada, que se pueden cuantificar.

Una CU hoja puede incluir una o más unidades de predicción (PU). En general, una PU representa un área espacial correspondiente a la totalidad o a una parte de la CU correspondiente, y puede incluir datos para recuperar una muestra de referencia para la PU. Por otra parte, una PU incluye datos relacionados con la predicción. Por ejemplo, cuando la PU está codificada en modo 'intra', los datos para la PU pueden estar incluidos en un árbol cuaternario residual (RQT), que puede incluir datos que describen un modo de intrapredicción para una TU correspondiente a la PU. Como otro ejemplo, cuando la PU está codificada en modo 'inter', la PU puede incluir datos que definen uno o más vectores de movimiento para la PU. Los datos que definen el vector de movimiento para una PU pueden describir, por ejemplo, una componente horizontal del vector de movimiento, una componente vertical del vector de movimiento, una resolución para el vector de movimiento (por ejemplo, precisión de un cuarto de píxel o precisión de un octavo de píxel), una imagen de referencia a la que apunta el vector de movimiento y/o una lista de imágenes de referencia (por ejemplo, Lista 0, Lista 1 o Lista C) para el vector de movimiento.

Una CU hoja que tiene una o más PU también puede incluir una o más unidades de transformada (TU). Las unidades de transformada se pueden especificar usando un RQT (también denominado estructura de árbol cuaternario de TU), como se analiza anteriormente. Por ejemplo, una bandera de división puede indicar si una CU hoja está dividida en cuatro unidades de transformada. A continuación, cada unidad de transformada se puede dividir en otras sub-TU. Cuando una TU no se divide más, se puede denominar TU hoja. En general, en la intracodificación, todas las TU hoja que pertenecen a una CU hoja comparten el mismo modo de intrapredicción. Es decir, el mismo modo de intrapredicción se aplica, en general, para calcular valores predichos para todas las TU de una CU hoja. En la intracodificación, un codificador de vídeo puede calcular un valor residual para cada TU hoja usando el modo de intrapredicción, como una diferencia entre la parte de la CU correspondiente a la TU y el bloque original. Una TU no se limita necesariamente al tamaño de una PU. Por tanto, las TU pueden ser más grandes o más pequeñas que una PU. En la intracodificación, una PU puede estar ubicada junto con una TU hoja correspondiente para la misma CU. En algunos ejemplos, el tamaño máximo de una TU hoja puede corresponder al tamaño de la CU hoja correspondiente.

Además, las TU de las CU hoja también pueden asociarse a respectivas estructuras de datos de árbol cuaternario, denominadas árboles cuaternarios residuales (RQT). Es decir, una CU hoja puede incluir un árbol cuaternario que indica cómo se divide la CU hoja en diversas TU. El nodo raíz de un árbol cuaternario de TU corresponde, en general, a una CU hoja, mientras que el nodo raíz de un árbol cuaternario de CU corresponde, en general, a un bloque de árbol (o LCU). Las TU del RQT que no están divididas se denominan TU hoja. En general, esta divulgación usa los términos CU y TU para referirse, respectivamente, a una CU hoja y una TU hoja, a menos que se indique lo contrario.

Una secuencia de vídeo incluye típicamente una serie de tramas o imágenes de vídeo. Un grupo de imágenes (GOP) comprende, en general, una serie de una o más de las imágenes de vídeo. Un GOP puede incluir, en una cabecera del GOP, una cabecera de una o más de las imágenes o en otro lugar, datos de sintaxis que describen una pluralidad de imágenes incluidas en el GOP. Cada segmento de una imagen puede incluir datos de sintaxis de segmento que describen un modo de codificación para el respectivo segmento. El codificador de vídeo 20 actúa típicamente en bloques de vídeo dentro de segmentos de vídeo individuales para codificar los datos de vídeo. Un bloque de vídeo puede corresponder a un nodo de codificación dentro de una CU. Los bloques de vídeo pueden tener tamaños fijos o variables, y pueden diferir en tamaño de acuerdo con una norma de codificación especificada.

Como ejemplo, el HM presta soporte a la predicción en varios tamaños de PU. Suponiendo que el tamaño de una CU en particular es  $2N \times 2N$ , el HM presta soporte a la intrapredicción en los tamaños de PU de  $2N \times 2N$  o  $N \times N$ , y a la interpredicción en los tamaños de PU simétricos de  $2N \times 2N$ ,  $2N \times N$ ,  $N \times 2N$  o  $N \times N$ . El HM también presta soporte a la división asimétrica para la interpredicción en los tamaños de PU de  $2N \times nU$ ,  $2N \times nD$ ,  $nL \times 2N$  y  $nR \times 2N$ . En la división asimétrica, una dirección de una CU no se divide, mientras que la otra dirección se divide en un 25 % y un 75 %. La parte de la CU correspondiente a la partición del 25 % se indica con una "n" seguida de una indicación de "encima" (U), "debajo" (D), "izquierda" (L) o "derecha" (R). Por tanto, por ejemplo, " $2N \times nU$ " se refiere a una CU  $2N \times 2N$  que está dividida horizontalmente con una PU  $2N \times 0,5N$  en la parte superior y una PU  $2N \times 1,5N$  en la parte inferior.

En esta divulgación, " $N \times N$ " y " $N$  por  $N$ " se pueden usar de manera intercambiable para referirse a las dimensiones en píxeles de un bloque de vídeo, en lo que respecta a dimensiones verticales y horizontales, por ejemplo,  $16 \times 16$  píxeles o 16 por 16 píxeles. En general, un bloque de tamaño  $16 \times 16$  tendrá 16 píxeles en una dirección vertical ( $y = 16$ ) y 16 píxeles en una dirección horizontal ( $x = 16$ ). Del mismo modo, un bloque  $N \times N$  tiene, en general,  $N$  píxeles en una dirección vertical y  $N$  píxeles en una dirección horizontal, donde  $N$  representa un valor entero no negativo. Los píxeles de un bloque pueden estar dispuestos en filas y columnas. Además, los bloques no han de tener necesariamente el mismo número de píxeles en la dirección horizontal que en la dirección vertical. Por ejemplo, los bloques pueden comprender  $N \times M$  píxeles, donde  $M$  no es necesariamente igual a  $N$ .

Después de la codificación intrapredictiva o interpredictiva usando las PU de una CU, el codificador de vídeo 20 puede calcular datos residuales para las TU de la CU. Las PU pueden comprender datos de sintaxis que describen un procedimiento o modo de generar datos de píxel predictivos en el dominio espacial (también denominado dominio de píxel) y las TU pueden comprender coeficientes en el dominio de transformada después de la aplicación de una transformada, por ejemplo, una transformada discreta de coseno (DCT), una transformada de enteros, una transformada de ondículas o una transformada conceptualmente similar, a datos de vídeo residuales. Los datos residuales pueden corresponder a diferencias de píxel entre píxeles de la imagen no codificada y valores de predicción correspondientes a las PU. El codificador de vídeo 20 puede formar las TU, incluidos los datos residuales para la CU y, a continuación, transformar las TU para producir coeficientes de transformada para la CU.

Después de cualquier transformada para producir coeficientes de transformada, el codificador de vídeo 20 puede realizar la cuantificación de los coeficientes de transformada. La cuantificación se refiere, en general, a un proceso en el que coeficientes de transformada se cuantifican para reducir posiblemente la cantidad de datos usados para representar los coeficientes, proporcionando una compresión adicional. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos o todos los coeficientes. Por ejemplo, un valor de  $n$  bits se puede redondear por defecto hasta un valor de  $m$  bits durante la cuantificación, donde  $n$  es mayor que  $m$ .

Después de la cuantificación, el codificador de vídeo puede explorar los coeficientes de transformada, generando un vector unidimensional a partir de la matriz bidimensional que incluye los coeficientes de transformada cuantificados.

La exploración se puede diseñar para que los coeficientes de mayor energía (y, por lo tanto, de menor frecuencia) se coloquen en la parte delantera de la matriz y para que los coeficientes de menor energía (y, por lo tanto, de mayor frecuencia) se coloquen en la parte posterior de la matriz. En algunos ejemplos, el codificador de vídeo 20 puede utilizar un orden de exploración predefinido para explorar los coeficientes de transformada cuantificados, para generar un vector serializado que se puede codificar entrópicamente. En otros ejemplos, el codificador de vídeo 20 puede realizar una exploración adaptativa. Después de explorar los coeficientes de transformada cuantificados para formar un vector unidimensional, el codificador de vídeo 20 puede codificar entrópicamente el vector unidimensional, por ejemplo, de acuerdo con la codificación de longitud variable adaptativa al contexto (CAVLC), la codificación aritmética binaria adaptativa al contexto (CABAC), la codificación aritmética binaria adaptativa al contexto basada en sintaxis (SBAC), la codificación entrópica por división en intervalos de probabilidad (PIPE) u otra metodología de codificación entrópica. El codificador de vídeo 20 también puede codificar entrópicamente elementos de sintaxis asociados a los datos de vídeo codificados para su uso por el descodificador de vídeo 30 en la descodificación de los datos de vídeo.

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

El codificador de vídeo 20 puede enviar además datos de sintaxis, tales como datos de sintaxis basados en bloque, datos de sintaxis basados en trama y datos de sintaxis basados en GOP, al descodificador de vídeo 30, por ejemplo, en una cabecera de trama, una cabecera de bloque, una cabecera de segmento o una cabecera de GOP. Los datos de sintaxis de GOP pueden describir un número de tramas en el GOP respectivo, y los datos de sintaxis de trama pueden indicar un modo de codificación/predicción usado para codificar la trama correspondiente.

En general, esta divulgación describe varios ejemplos ilustrativos de soluciones para permitir una ampliación únicamente de sintaxis de alto nivel (HLS) de una norma de codificación de vídeo, tal como HEVC. Por ejemplo, estas técnicas se pueden usar para desarrollar una ampliación solamente de HLS para un perfil de HEVC, tal como MVC o SVC. A continuación se describen diversos ejemplos. Debe entenderse que aunque se describen diversos ejemplos por separado, los elementos de cualquiera de los ejemplos o de todos ellos pueden combinarse en cualquier combinación. También debe entenderse que la invención reivindicada comprende inhabilitar la predicción de vectores de movimiento entre imágenes de referencia a corto y largo plazo, y que cualquier combinación de este tipo que no comprenda la inhabilitación de la predicción de vectores de movimiento entre imágenes de referencia a corto y largo plazo no está dentro del alcance de la invención reivindicada.

En un primer ejemplo, no hay cambios en la especificación base de HEVC actual. En la ampliación de HEVC, una imagen (por ejemplo, una componente de vista) puede identificarse mediante dos propiedades: su valor de recuento de orden de imagen (POC) y un identificador de imagen de segunda dimensión, por ejemplo, un valor de view\_id (que puede identificar una vista en la que la imagen está presente). Puede ser necesario que el codificador de vídeo 20 indique una componente de vista que se usará en la predicción entre vistas como una imagen de referencia a largo plazo.

En un segundo ejemplo, no hay cambios en la especificación base de HEVC actual. En la ampliación de HEVC se pueden aplicar los siguientes cambios. Una imagen (por ejemplo, una componente de vista) puede identificarse mediante dos propiedades: un valor de POC y un identificador de imagen de segunda dimensión, por ejemplo, view\_id. En este segundo ejemplo, un proceso de marcado de imagen adicional puede introducirse inmediatamente antes de codificar una componente de vista actual para marcar todas las imágenes de referencia entre vistas como imágenes de referencia a largo plazo. Otro proceso de marcado de imágenes se puede introducir inmediatamente después de codificar una componente de vista actual para marcar cada imagen de referencia entre vistas como a largo plazo, a corto plazo o "no usada como referencia", que es lo mismo que su estado de marcado previo antes de codificar la componente de vista actual.

En un tercer ejemplo, las técnicas del segundo ejemplo se usan y complementan como sigue. Además de las técnicas del segundo ejemplo, para cada imagen de referencia entre vistas, después de marcarse como una imagen de referencia a largo plazo, su valor de POC se mapea a un nuevo valor de POC, que no es equivalente al valor de POC de cualquier imagen de referencia existente. Después de descodificar la componente de vista actual, para cada imagen de referencia entre vistas, su valor de POC se mapea de nuevo al valor de POC original, que es igual a la componente de vista actual. Por ejemplo, la componente de vista actual puede pertenecer a la vista 3 (suponiendo que el identificador de vista es igual al índice de orden de vista) y puede tener un valor de POC igual a 5. Dos imágenes de referencia entre vistas pueden tener sus valores de POC (que son ambos 5) convertidos a, por ejemplo, 1025 y 2053. Después de descodificar la componente de vista actual, los valores de POC de las imágenes entre vistas se pueden convertir de nuevo a 5.



En un cuarto ejemplo, y de acuerdo con la invención, las técnicas del segundo o tercer ejemplo se pueden usar y complementar como sigue. Además de las técnicas del primer ejemplo o del segundo ejemplo, como se mencionó anteriormente, en la especificación base de HEVC, se usa un gancho adicional para inhabilitar la predicción entre cualquier vector de movimiento que se refiera a una imagen a corto plazo y otro vector de movimiento que se refiera a imágenes a largo plazo, especialmente durante la predicción avanzada de vectores de movimiento (AMVP).

En un quinto ejemplo, en la ampliación de HEVC, una imagen puede identificarse mediante dos propiedades: un valor de POC y una identificación de imagen de segunda dimensión, por ejemplo, `view_id`. En la especificación base de HEVC, se pueden añadir uno o más de los siguientes ganchos (solos o en cualquier combinación). En un ejemplo (al que se hace referencia como ejemplo 5.1), cuando se identifica una imagen de referencia durante AMVP y el modo de fusión, se puede usar una identificación de imagen de segunda dimensión, por ejemplo, índice de orden de visualización, junto con POC. En el contexto de la descodificación de vídeo bidimensional 2D en la especificación base de HEVC, la identificación de imagen de segunda dimensión siempre puede establecerse igual a 0.

En otro ejemplo (ejemplo 5.2), la predicción entre un vector de movimiento temporal y un vector de movimiento entre vistas está inhabilitada durante AMVP (incluida la predicción de vectores de movimiento temporal (TMVP)). Una propiedad del vector de movimiento puede decidirse mediante el índice de referencia asociado, que identifica una imagen de referencia y la manera en que la imagen de referencia está siendo referenciada por la imagen que contiene el vector de movimiento, por ejemplo, como una imagen de referencia a largo plazo, una imagen de referencia a corto plazo o una imagen de referencia entre vistas. En otro ejemplo (ejemplo 5.3), y de acuerdo con la invención, la predicción entre un vector de movimiento temporal a corto plazo y un vector de movimiento temporal a largo plazo está inhabilitada (por ejemplo, explícita o implícitamente). En otro ejemplo (ejemplo 5.4), la predicción entre un vector de movimiento temporal a corto plazo y un vector de movimiento temporal a largo plazo puede estar habilitada (por ejemplo, explícita o implícitamente).

En otro ejemplo (ejemplo 5.5), la predicción entre vectores de movimiento que hacen referencia a dos imágenes de referencia diferentes entre vistas puede estar inhabilitada (por ejemplo, explícita o implícitamente). Se puede considerar que dos imágenes de referencia entre vistas tienen tipos diferentes si los valores de identificador de imagen de segunda dimensión para las mismas son diferentes. En otro ejemplo (ejemplo 5.6), la predicción entre vectores de movimiento que hacen referencia a dos imágenes de referencia diferentes entre vistas puede estar habilitada (por ejemplo, explícita o implícitamente). En otro ejemplo (ejemplo 5.7), la predicción entre vectores de movimiento que hacen referencia a una imagen a largo plazo y a un modo entre vistas puede estar habilitada (por ejemplo, explícita o implícitamente). En otro ejemplo (ejemplo 5.8), la predicción entre vectores de movimiento que hacen referencia a una imagen a largo plazo y a un modo entre vistas puede estar inhabilitada (por ejemplo, explícita o implícitamente).

En cualquiera de los ejemplos anteriores, siempre se puede habilitar la predicción entre dos vectores de movimiento que hacen referencia a dos imágenes de referencia temporales a corto plazo diferentes y se puede habilitar el escalado de una a otra en base a los valores de POC. De forma adicional o alternativa, en cualquiera de los ejemplos anteriores, la predicción entre vectores de movimiento que hacen referencia a dos imágenes a largo plazo diferentes puede estar inhabilitada. Determinados detalles de los diversos ejemplos descritos anteriormente se analizan a continuación con mayor detalle.

En general, esta divulgación se refiere a un "vector de movimiento" o "datos de vector de movimiento" que incluye(n) un índice de referencia (es decir, un puntero a una imagen de referencia) y coordenadas x-y del propio vector de movimiento. Tanto un vector de movimiento de disparidad como un vector de movimiento temporal pueden denominarse, en general, "vectores de movimiento". Una imagen de referencia correspondiente a un índice de referencia puede denominarse imagen de referencia a la que hace referencia un vector de movimiento. Si un vector de movimiento hace referencia a una imagen de referencia en la misma vista, se denomina vector de movimiento temporal. Si un vector de movimiento hace referencia a una imagen de referencia de una vista diferente, se denomina vector de movimiento de disparidad.

Un vector de movimiento temporal puede ser un vector de movimiento temporal a corto plazo ("vector de movimiento a corto plazo") o un vector de movimiento temporal a largo plazo ("vector de movimiento a largo plazo"). Por ejemplo, un vector de movimiento es a corto plazo si hace referencia a una imagen de referencia a corto plazo, mientras que un vector de movimiento es a largo plazo si hace referencia a una imagen de referencia a largo plazo. Cabe señalar que, a menos que se mencione lo contrario, un vector de movimiento de disparidad y un vector de movimiento a largo plazo describen, en general, diferentes categorías de vectores de movimiento, por ejemplo, para la predicción entre vistas y la predicción temporal entre vistas, respectivamente. Las imágenes de referencia a corto y largo plazo representan ejemplos de imágenes de referencia temporales.

El codificador de vídeo 20 y el descodificador de vídeo 30 pueden configurarse para identificar una imagen de referencia de un búfer de imágenes descodificadas (DPB), que puede implementarse como una memoria de imágenes de referencia. El proceso de identificación de una imagen de referencia del DPB se puede usar en cualquiera de los ejemplos de las técnicas descritas en esta divulgación. El proceso de identificación de una imagen de referencia del DPB se puede usar para los siguientes propósitos en la especificación de la ampliación de HEVC: generación de

conjunto de imágenes de referencia, generación de lista de imágenes de referencia y/o marcado de imágenes de referencia.

Una componente de vista, una componente de vista de textura, una componente de vista de profundidad o una capa escalable (con, por ejemplo, una combinación específica de `dependency_id` y `quality_id`) pueden identificarse con un valor de recuento de orden de imagen (POC) y una información de identificación de imagen de segunda dimensión. La información de identificación de imagen de segunda dimensión puede incluir uno o más de lo siguiente: ID de vista (`view_id`) en contexto multivista; índice de orden de vista en contexto multivista; en contexto de 3DV (multivista con profundidad), una combinación de índice de orden de vista y una bandera de profundidad (que indica si la componente de vista actual es textura o profundidad), por ejemplo, índice de orden de vista multiplicado por dos más el valor de la bandera de profundidad (`depth_flag`); en el contexto de SVC, ID de capa (en un entorno de codificación escalable, por ejemplo, en SVC basada en AVC, el ID de capa puede ser igual a `dependency_id` multiplicado por 16 más `quality_id`); o un ID de capa genérico (`layer_id`), por ejemplo, el valor de `reserved_one_5bits` menos 1, en el que `reserved_one_5bits` es como se especifica en la especificación base de HEVC. Cabe señalar que una ID de capa genérica puede ser aplicable a escenarios mixtos de 3DV (multivista con profundidad) y escalabilidad. Los ejemplos mencionados anteriormente pueden aplicarse a cualquier códec de múltiples capas, incluido el códec de vídeo escalable, por ejemplo, considerando cada capa como una vista. En otras palabras, para la codificación de vídeo multivista, las diversas vistas pueden considerarse capas separadas.

En algunos escenarios, una capa base o una vista dependiente puede tener múltiples representaciones, por ejemplo, debido al uso de diferentes filtros de muestreo ascendente/suavizado, o debido al hecho de usar una imagen sintetizada de vista para la predicción; por lo tanto, en una ubicación de vista, puede haber dos imágenes listas para usar, donde una es la imagen de vista dependiente reconstruida normal y la otra es la imagen de vista sintetizada, ambas con el mismo `view_id` o índice de orden de vista. En este caso, se puede usar una identificación de imagen de tercera dimensión.

El codificador de vídeo 20 y el decodificador de vídeo 30 también pueden configurarse para identificar una imagen de referencia a partir de las listas de imágenes de referencia. El búfer de imágenes descodificadas (DPB) se puede organizar en listas de imágenes de referencia, por ejemplo, `RefPicList0`, que incluye posibles imágenes de referencia que tienen valores de POC menores que el valor de POC de una imagen actual, y `RefPicList1`, que incluye posibles imágenes de referencia que tienen valores de POC mayores que el valor de POC de la imagen actual. Las técnicas para identificar una imagen de referencia a partir de una lista de imágenes de referencia se usan como gancho para la especificación base actual de HEVC. Las funciones definidas pueden ser invocadas varias veces por un codificador de vídeo o un decodificador de vídeo durante la AMVP y el modo de fusión.

Una componente de vista, una componente de vista de textura, una componente de vista de profundidad o una capa escalable (con, por ejemplo, una combinación específica de `dependency_id` y `quality_id`) se pueden identificar con un valor de POC y una información de identificación de imagen de segunda dimensión, que puede ser uno de lo siguiente: índice de orden de vista en el contexto de multivista o 3DV. La función `viewOldx(pic)` devuelve el índice de orden de vista de la vista a la que pertenece la imagen identificada como "pic". Esta función devuelve 0 para cualquier componente de vista, componente de vista de textura o componente de vista de profundidad de la vista base; ID de vista (`view_id`); en el contexto 3DV, una combinación de índice de orden de vista y una bandera de profundidad (que indica si la componente de vista actual es textura o profundidad); índice de orden de vista multiplicado por dos más el valor de `depth_flag`; en el contexto de SVC, ID de capa (en un entorno de codificación escalable, por ejemplo, en SVC basada en AVC, el ID de capa puede ser igual a `dependency_id` multiplicado por 16 más `quality_id`); o un ID de capa genérico (`layer_id`), por ejemplo, el valor de `reserved_one_5bits` menos 1, en el que `reserved_one_5bits` es como se especifica en la especificación base de HEVC. La función `layerId(pic)` devuelve el ID de capa de la imagen pic. `LayerId(pic)` devuelve 0 para cualquier componente de vista (textura) de la vista base. `LayerId(pic)` devuelve 0 para cualquier imagen (o representación de capa) de la capa base de SVC. Cabe señalar que una ID de capa genérica puede ser aplicable a escenarios mixtos de 3DV (multivista con profundidad) y escalabilidad.

En algunos escenarios, una capa base o una vista dependiente puede tener múltiples representaciones, por ejemplo, debido al uso de diferentes filtros de muestreo ascendente/suavizado, o debido al hecho de usar una imagen sintetizada de vista para la predicción; por lo tanto, en una ubicación de vista, puede haber dos imágenes listas para usar: una es la imagen de vista dependiente reconstruida normal, la otra es la imagen de vista sintetizada, ambas con el mismo `view_id` o índice de orden de vista. En este caso, se puede usar una identificación de imagen de tercera dimensión.

Una o más de las identificaciones de imagen de segunda y/o tercera dimensión mencionadas anteriormente pueden definirse usando la función `AddPicId(pic)`.

El codificador de vídeo 20 y el decodificador de vídeo 30 también pueden configurarse para identificar un tipo de entrada en una lista de imágenes de referencia. Esto se puede usar como gancho para la especificación base actual de HEVC. Cualquiera o todas las funciones definidas a continuación pueden invocarse varias veces mediante el codificador de vídeo 20 y/o el decodificador de vídeo 30 durante la AMVP y/o el modo de fusión. Cualquiera o todas las técnicas de ejemplo siguientes, donde no todas pertenecen a la invención, se pueden usar para identificar el tipo

de una entrada en una lista de imágenes de referencia. En un ejemplo que no pertenece a la invención, una función "RefPicType(pic)" devuelve 0 si la imagen pic es una imagen de referencia temporal, y devuelve 1 si la imagen pic no es una imagen de referencia temporal. En otro ejemplo que no pertenece a la invención, una función RefPicType(pic) devuelve 0 si la imagen pic tiene el mismo POC que la imagen actual, y devuelve 1 si la imagen pic tiene un POC diferente al de la imagen actual.

En otro ejemplo de acuerdo con la invención, los resultados de los ejemplos analizados anteriormente se pueden lograr reemplazando el uso de la función RefPicType(pic) simplemente verificando si el POC de "pic" (el argumento de la función) es igual al POC de la imagen actual. Una imagen de referencia entre vistas puede marcarse como "no usada como referencia". Para simplificar, dicha imagen se denomina imagen de no referencia en la especificación base de HEVC. En algunos ejemplos, una imagen marcada como "usada como referencia a largo plazo" o "usada como referencia a corto plazo" puede denominarse imagen de referencia en la especificación base de HEVC. En algunos ejemplos que no pertenecen a la invención, la función RefPicType(pic) devuelve 0 si la imagen pic está marcada como "usada como referencia a largo plazo" o "usada como referencia a corto plazo" y devuelve 1 si la imagen pic está marcada como "no usada como referencia". Además, en algunos ejemplos, en la ampliación de HEVC, una componente de vista, inmediatamente después de su descodificación, puede marcarse como "no usada como referencia", independientemente del valor del elemento de sintaxis nal\_ref\_flag.

Después de codificar toda la unidad de acceso, las componentes de vista de la unidad de acceso pueden marcarse como "usadas como referencia a corto plazo" o "usadas como referencia a largo plazo" si nal\_ref\_flag es verdadero. De forma alternativa, una componente de vista solo puede marcarse como "usada como referencia a corto plazo" o "usada como referencia a largo plazo" si está incluida en el conjunto de imágenes de referencia (RPS) de una componente de vista subsiguiente en el orden de descodificación en la misma vista, inmediatamente después de que se obtenga el RPS para la componente de vista subsiguiente. Además, en la especificación base de HEVC, una imagen actual, inmediatamente después de su descodificación, puede marcarse como "no usada como referencia".

En algunos ejemplos, RefPicType(picX, refIdx, LX) devuelve el valor de RefPicType(pic) en el momento en que picX era la imagen actual, donde pic es la imagen de referencia con índice refIdx de la lista de imágenes de referencia LX de la imagen picX.

Con respecto al ejemplo mencionado anteriormente como el "cuarto ejemplo", el codificador de vídeo 20 y el descodificador de vídeo 30 pueden configurarse para permitir la predicción entre imágenes de referencia a largo plazo sin escalado durante la AMVP y la TMVP. Con respecto a AMVP, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden configurarse para realizar un proceso de obtención modificado para candidatos a predictor de vectores de movimiento (MVP). Los datos de entrada para el proceso pueden incluir una ubicación de luma (xP, yP) que especifica la muestra de luma superior izquierda de la unidad de predicción actual en relación con la muestra superior izquierda de la imagen actual, variables que especifican la anchura y la altura de la unidad de predicción para luma, nPSW y nPSH, y el índice de referencia de la partición de unidad de predicción actual refIdxLX (siendo X 0 o 1). Los datos de salida del proceso pueden incluir (donde N se reemplaza por A o B, donde A corresponde a candidatos vecinos situados a la izquierda y B corresponde a candidatos vecinos situados por encima, como se muestra en el ejemplo de la FIG.10) los vectores de movimiento mvLXN de las unidades de predicción vecinas y las banderas de disponibilidad availableFlagLXN de las unidades de predicción vecinas. La variable isScaledFlagLX, donde X es 0 o 1, puede establecerse igual a 0.

El codificador de vídeo 20 y el descodificador de vídeo 30 pueden obtener el vector de movimiento mvLXA y la bandera de disponibilidad availableFlagLXA en las siguientes etapas ordenadas en un ejemplo, donde el texto subrayado representa cambios con respecto a HEVC WD7:

1. Sea un conjunto de dos ubicaciones de muestra ( $x_{A_k}$ ,  $y_{A_k}$ ), con  $k = 0, 1$ , que especifica ubicaciones de muestra con  $x_{A_k} = x_P - 1$ ,  $y_{A_0} = y_P + nPSH$  e  $y_{A_1} = y_{A_0} - \text{MinPuSize}$ . El conjunto de ubicaciones de muestra ( $x_{A_k}$ ,  $y_{A_k}$ ) representa las ubicaciones de muestra justo en el lado izquierdo del límite de partición izquierdo y su línea extendida.

2. Establézcase la bandera de disponibilidad availableFlagLXA inicialmente igual 0 y ambas componentes de mvLXA igual a 0.

3. Cuando una o más de las siguientes condiciones son verdaderas, la variable isScaledFlagLX se establece igual a 1, en este ejemplo.

- La unidad de predicción que cubre la ubicación de luma ( $x_{A_0}$ ,  $y_{A_0}$ ) está disponible [Ed. (BB): se reescribe usando MinCbAddrZS[ ][ ] y el proceso de disponibilidad para bloques de codificación mínimos] y PredMode no es MODE\_INTRA.

- La unidad de predicción que cubre la ubicación de luma ( $x_{A_1}$ ,  $y_{A_1}$ ) está disponible [Ed. (BB): se reescribe usando MinCbAddrZS[ ][ ] y el proceso de disponibilidad para bloques de codificación mínimos] y PredMode no es MODE\_INTRA.

4. Para  $(x_{A_k}, y_{A_k})$  desde  $(x_{A_0}, y_{A_0})$  hasta  $(x_{A_1}, y_{A_1})$  donde  $y_{A_1} = y_{A_0} - \text{MinPuSize}$ , lo siguiente se aplica repetidamente hasta que `availableFlagLXA` sea igual a 1:

- Si la unidad de predicción que cubre la ubicación de luma  $(x_{A_k}, y_{A_k})$  está disponible [Ed. (BB): se reescribe usando `MinCbAddrZS[ ][ ]` y el proceso de disponibilidad para bloques de codificación mínimos], `PredMode` no es `MODE_INTRA`, `predFlagLX[xAk][yAk]` es igual a 1 y el índice de referencia `refldxLX[xAk][yAk]` es igual al índice de referencia de la unidad de predicción actual `refldxLX`, `availableFlagLXA` se establece igual a 1 y el vector de movimiento `mvLXA` se establece igual al vector de movimiento `mvLX[xAk][yAk]`, `refldxA` se establece igual a `refldxLX[xAk][yAk]` y `ListA` se establece igual a `ListX`.

- De lo contrario, si la unidad de predicción que cubre la ubicación de luma  $(x_{A_k}, y_{A_k})$  está disponible [Ed. (BB): se reescribe usando `MinCbAddrZS[ ][ ]` y el proceso de disponibilidad para bloques de codificación mínimos], `PredMode` no es `MODE_INTRA`, `predFlagLY[xAk][yAk]` (con  $Y \neq X$ ) es igual a 1 y `PicOrderCnt(RefPicListY[refldxLY[xAk][yAk]])` es igual a `PicOrderCnt(RefPicListX[refldxLX])`, `availableFlagLXA` se establece igual a 1, el vector de movimiento `mvLXA` se establece igual al vector de movimiento `mvLY[xAk][yAk]`, `refldxA` se establece igual a `refldxLY[xAk][yAk]`, `ListA` se establece igual a `ListY` y `mvLXA` se establece igual a `mvLY`.

5. Cuando `availableFlagLXA` es igual a 0, para  $(x_{A_k}, y_{A_k})$  desde  $(x_{A_0}, y_{A_0})$  hasta  $(x_{A_1}, y_{A_1})$  donde  $y_{A_1} = y_{A_0} - \text{MinPuSize}$ , lo siguiente se aplica repetidamente hasta que `availableFlagLXA` sea igual a 1, en este ejemplo:

- Si la unidad de predicción que cubre la ubicación de luma  $(x_{A_k}, y_{A_k})$  está disponible [Ed. (BB): se reescribe usando `MinCbAddrZS[ ][ ]` y el proceso de disponibilidad para bloques de codificación mínimos], `PredMode` no es `MODE_INTRA`, `predFlagLX[xAk][yAk]` es igual a 1 y `RefPicListX[refldxLX]` y `RefPicListX[refldxLX[xAk][yAk]]` son ambas imágenes de referencia a largo plazo o son ambas imágenes de referencia a corto plazo, `availableFlagLXA` se establece igual a 1, el vector de movimiento `mvLXA` se establece igual al vector de movimiento `mvLX[xAk][yAk]`, `refldxA` se establece igual a `refldxLX[xAk][yAk]`, `ListA` se establece igual a `ListX`.

- De lo contrario, si la unidad de predicción que cubre la ubicación de luma  $(x_{A_k}, y_{A_k})$  está disponible [Ed. (BB): se reescribe usando `MinCbAddrZS[ ][ ]` y el proceso de disponibilidad para bloques de codificación mínimos], `PredMode` no es `MODE_INTRA`, `predFlagLY[xAk][yAk]` (con  $Y \neq X$ ) es igual a 1, y `RefPicListX[refldxLX]` y `RefPicListY[refldxLY[xAk][yAk]]` son ambas imágenes de referencia a largo plazo o son ambas imágenes de referencia a corto plazo, `availableFlagLXA` se establece igual a 1, el vector de movimiento `mvLXA` se establece igual al vector de movimiento `mvLY[xAk][yAk]`, `refldxA` se establece igual a `refldxLY[xAk][yAk]`, `ListA` se establece igual a `ListY`.

- Cuando `availableFlagLXA` es igual a 1, y tanto `RefPicListA[refldxA]` como `RefPicListX[refldxLX]` son imágenes de referencia a corto plazo, `mvLXA` se obtiene como se especifica a continuación (donde la notación 8-### se refiere a secciones del borrador actual de HEVC, es decir, WD7).

$$tx = (16384 + (\text{Abs}(td) \gg 1)) / td \quad (8-126)$$

$$\text{DistScaleFactor} = \text{Clip3}(-4096, 4095, (tb * tx) \gg 6) \quad (8-127)$$

$$\begin{aligned} mvLXA &= \text{Clip3}(-8192, 8191.75, \text{Sign}(\text{DistScaleFactor} * mvLXA) * \\ &((\text{Abs}(\text{DistScaleFactor} * mvLXA) + 127) \gg 8)) \end{aligned} \quad (8-128)$$

donde `td` y `tb` pueden obtenerse como

$$td = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListA}[\text{refldxA}])) \quad (8-129)$$

$$tb = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListA}[\text{refldxA}])) \quad (8-130)$$

El codificador de vídeo 20 y el descodificador de vídeo 30 pueden configurarse para obtener el vector de movimiento `mvLXB` y la bandera de disponibilidad `availableFlagLXB` en las siguientes etapas ordenadas en un ejemplo, donde el texto subrayado representa cambios con respecto a HEVC WD7:

1. Sea un conjunto de tres ubicaciones de muestra  $(x_{B_k}, y_{B_k})$ , con  $k = 0, 1, 2$ , que especifican ubicaciones de muestra con  $x_{B_0} = x_P + nPSW$ ,  $x_{B_1} = x_{B_0} - \text{MinPuSize}$ ,  $x_{B_2} = x_P - \text{MinPuSize}$  e  $y_{B_k} = y_P - 1$ . El conjunto de ubicaciones de muestra  $(x_{B_k}, y_{B_k})$  representa las ubicaciones de muestra justo en el lado superior del límite de partición superior y su línea extendida. [Ed. (BB): se define `MinPuSize` en el SPS, pero la obtención debe depender del uso de una bandera `AMP`]

2. Cuando  $y_P - 1$  es menor que  $((y_C \gg \text{Log2CtbSize}) \ll \text{Log2CtbSize})$ , se aplica lo siguiente.

$$xB_0 = (xB_0 \gg 3) \ll 3 + ((xB_0 \gg 3) \& 1) * 7 \quad (8-131)$$

$$xB_1 = (xB_1 \gg 3) \ll 3 + ((xB_1 \gg 3) \& 1) * 7 \quad (8-132)$$

$$xB_2 = (xB_2 \gg 3) \ll 3 + ((xB_2 \gg 3) \& 1) * 7 \quad (8-133)$$

3. Establézcase la bandera de disponibilidad availableFlagLXB inicialmente igual 0 y ambas componentes de mvLXB igual a 0.

4. Para  $(xB_k, yB_k)$  desde  $(xB_0, yB_0)$  hasta  $(xB_2, yB_2)$  donde  $xB_0 = xP + nPSW$ ,  $xB_1 = xB_0 - MinPuSize$ , y  $xB_2 = xP - MinPuSize$ , lo siguiente se aplica repetidamente hasta que availableFlagLXB sea igual a 1:

- Si la unidad de predicción que cubre la ubicación de luma  $(xB_k, yB_k)$  está disponible [Ed. (BB): se reescribe usando MinCbAddrZS[ ] y el proceso de disponibilidad para bloques de codificación mínimos], PredMode no es MODE\_INTRA, predFlagLX[ $xB_k$ ][ $yB_k$ ] es igual a 1 y el índice de referencia refIdxLX[ $xB_k$ ][ $yB_k$ ] es igual al índice de referencia de la unidad de predicción actual refIdxLX, availableFlagLXB se establece igual a 1 y el vector de movimiento mvLXB se establece igual al vector de movimiento mvLX[ $xB_k$ ][ $yB_k$ ], refIdxB se establece igual a refIdxLX[ $xB_k$ ][ $yB_k$ ] y ListB se establece igual a ListX.

- De lo contrario, si la unidad de predicción que cubre la ubicación de luma  $(xB_k, yB_k)$  está disponible [Ed. (BB): se reescribe usando MinCbAddrZS[ ] y el proceso de disponibilidad para bloques de codificación mínimos], PredMode no es MODE\_INTRA, predFlagLY[ $xB_k$ ][ $yB_k$ ] (con  $Y \neq X$ ) es igual a 1 y PicOrderCnt(RefPicListY[refIdxLY[ $xB_k$ ][ $yB_k$ ]]) es igual a PicOrderCnt(RefPicListX[refIdxLX]), availableFlagLXB se establece igual a 1, el vector de movimiento mvLXB se establece igual al vector de movimiento mvLY[ $xB_k$ ][ $yB_k$ ], refIdxB se establece igual a refIdxLY[ $xB_k$ ][ $yB_k$ ], y ListB se establece igual a ListY.

5. Cuando isScaledFlagLX es igual a 0 y availableFlagLXB es igual a 1, mvLXA se establece igual a mvLXB y refIdxA se establece igual a refIdxB y availableFlagLXA se establece igual a 1.

6. Cuando isScaledFlagLX es igual a 0, availableFlagLXB se establece igual a 0 y para  $(xB_k, yB_k)$  desde  $(xB_0, yB_0)$  hasta  $(xB_2, yB_2)$  donde  $xB_0 = xP + nPSW$ ,  $xB_1 = xB_0 - MinPuSize$ , y  $xB_2 = xP - MinPuSize$ , lo siguiente se aplica repetidamente hasta que availableFlagLXB sea igual a 1:

- Si la unidad de predicción que cubre la ubicación de luma  $(xB_k, yB_k)$  está disponible [Ed. (BB): se reescribe usando MinCbAddrZS[ ] y el proceso de disponibilidad para bloques de codificación mínimos], PredMode no es MODE\_INTRA, predFlagLX[ $xB_k$ ][ $yB_k$ ] es igual a 1 y RefPicListX[refIdxLX] y RefPicListX[refIdxLX[ $xB_k$ ][ $yB_k$ ]] son ambas imágenes de referencia a largo plazo o son ambas imágenes de referencia a corto plazo, availableFlagLXB se establece igual a 1, el vector de movimiento mvLXB se establece igual al vector de movimiento mvLX[ $xB_k$ ][ $yB_k$ ], refIdxB se establece igual a refIdxLX[ $xB_k$ ][ $yB_k$ ], ListB se establece igual a ListX.

- De lo contrario, si la unidad de predicción que cubre la ubicación de luma  $(xB_k, yB_k)$  está disponible [Ed. (BB): se reescribe usando MinCbAddrZS[ ] y el proceso de disponibilidad para bloques de codificación mínimos], PredMode no es MODE\_INTRA, predFlagLY[ $xB_k$ ][ $yB_k$ ] (con  $Y \neq X$ ) es igual a 1, y RefPicListX[refIdxLX] y RefPicListY[refIdxLY[ $xB_k$ ][ $yB_k$ ]] son ambas imágenes de referencia a largo plazo o son ambas imágenes de referencia a corto plazo, availableFlagLXB se establece igual a 1, el vector de movimiento mvLXB se establece igual al vector de movimiento mvLY[ $xB_k$ ][ $yB_k$ ], refIdxB se establece igual a refIdxLY[ $xB_k$ ][ $yB_k$ ], ListB se establece igual a ListY.

- Cuando availableFlagLXB es igual a 1 y PicOrderCnt(RefPicListB[refIdxB]) no es igual a PicOrderCnt(RefPicListX[refIdxLX]) y tanto RefPicListB[refIdxB] como RefPicListX[refIdxLX] son imágenes de referencia de corto plazo, mvLXB puede obtenerse como se especifica a continuación (donde la notación 8-### se refiere a secciones del borrador actual de HEVC).

$$tx = (16384 + (Abs(td) \gg 1)) / td \quad (8-134)$$

$$DistScaleFactor = Clip3(-4096, 4095, (tb * tx) \gg 6) \quad (8-135)$$

$$mvLXB = Clip3(-8192, 8191.75, Sign(DistScaleFactor * mvLXA) * ((Abs(DistScaleFactor * mvLXA) + 127) \gg 8)) \quad (8-136)$$

donde td y tb pueden obtenerse como

$$td = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListB}[\text{refIdxB}])) \quad (8-137)$$

$$td = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListX}[\text{refIdxLX}])) \quad (8-138)$$

El codificador de vídeo 20 y el decodificador de vídeo 30 también pueden configurarse para realizar un proceso de obtención modificado para la predicción de vectores de movimiento temporal (TMVP) para codificar vectores de movimiento de bloques de luminancia. En un ejemplo, los datos de entrada para este proceso pueden incluir una ubicación de luma (xP, yP) que especifica la muestra de luma superior izquierda de la unidad de predicción actual en relación con la muestra superior izquierda de la imagen actual, variables que especifican la anchura y la altura de la unidad de predicción para luma, nPSW y nPSH, y el índice de referencia de la partición de unidad de predicción actual refIdxLX (siendo X 0 o 1). Los datos de salida de este proceso pueden incluir la predicción de vectores de movimiento mvLXCol y la bandera de disponibilidad availableFlagLXCol.

En un ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden ejecutar una función  $\text{RefPicOrderCnt}(\text{picX}, \text{refIdx}, \text{LX})$  que devuelve el recuento de orden de imagen  $\text{PicOrderCntVal}$  de la imagen de referencia con índice  $\text{refIdx}$  de la lista de imágenes de referencia LX de la imagen  $\text{picX}$ . Esta función se puede especificar como sigue, donde (8-141) y referencias similares en esta descripción hacen referencia a secciones de HEVC WD7:

$$\text{RefPicOrderCnt}(\text{picX}, \text{refIdx}, \text{LX}) = \text{PicOrderCnt}(\text{RefPicListX}[\text{refIdx}] \text{ de la imagen } \text{picX}) \quad (8-141)$$

Dependiendo de los valores de  $\text{slice\_type}$ ,  $\text{collocated\_from\_10\_flag}$  y  $\text{collocated\_ref\_idx}$ , la variable  $\text{colPic}$ , que especifica la imagen que contiene la partición coubicada, puede obtenerse como sigue:

- Si el tipo de segmento es igual a B y  $\text{collocated\_from\_10\_flag}$  es igual a 0, la variable  $\text{colPic}$  especifica la imagen que contiene la partición coubicada según lo especificado por  $\text{RefPicList1}[\text{collocated\_ref\_idx}]$ .

- De lo contrario ( $\text{slice\_type}$  es igual a B y  $\text{collocated\_from\_10\_flag}$  es igual a 1 o  $\text{slice\_type}$  es igual a P), la variable  $\text{colPic}$  especifica la imagen que contiene la partición coubicada según lo especificado por  $\text{RefPicList0}[\text{collocated\_ref\_idx}]$ .

La variable  $\text{colPu}$  y su posición ( $\text{xPCol}$ ,  $\text{yPCol}$ ) se pueden obtener usando las siguientes etapas ordenadas:

1. La variable  $\text{colPu}$  se puede obtener como sigue

$$yPRb = yP + nPSH \quad (8-139)$$

- Si ( $yP \gg \text{Log2CtbSize}$ ) es igual a ( $yPRb \gg \text{Log2CtbSize}$ ), la componente horizontal de la posición de luma inferior derecha de la unidad de predicción actual se define mediante

$$xPRb = xP + nPSW \quad (8-140)$$

y la variable  $\text{colPu}$  se establece como la unidad de predicción que cubre la posición modificada dada por  $((xPRb \gg 4) \ll 4, (yPRb \gg 4) \ll 4)$  dentro de  $\text{colPic}$ .

- De lo contrario ( $(yP \gg \text{Log2CtbSize})$  no es igual a  $(yPRb \gg \text{Log2CtbSize})$ ),  $\text{colPu}$  se marca como "no disponible".

2. Cuando  $\text{colPu}$  se codifica en un modo de intrapredicción o  $\text{colPu}$  se marca como "no disponible", se aplica lo siguiente.

- La posición de luma central de la unidad de predicción actual se define mediante

$$xPCtr = (xP + nPSW \gg 1) \quad (8-141)$$

$$yPCtr = (yP + nPSH \gg 1) \quad (8-142)$$

- La variable  $\text{colPu}$  se establece como la unidad de predicción que cubre la posición modificada dada por  $((xPCtr \gg 4) \ll 4, (yPCtr \gg 4) \ll 4)$  dentro de  $\text{colPic}$ .

( $\text{xPCol}$ ,  $\text{yPCol}$ ) se establece igual a la muestra de luma superior izquierda de  $\text{colPu}$  en relación con la muestra de luma superior izquierda de  $\text{colPic}$ .

La función LongTermRefPic(picX, reldx, LX) se puede definir como sigue. Si la imagen de referencia con índice reldx de la lista de imágenes de referencia LX de la imagen picX se marcó como "usada como referencia a largo plazo" en el momento en que picX era la imagen actual, LongTermRefPic(picX, reldx, LX) devuelve 1; de lo contrario, LongTermRefPic(picX, reldx, LX) devuelve 0.

Las variables mvLXCol y availableFlagLXCol pueden obtenerse como sigue, donde el texto subrayado representa cambios con respecto a HEVC WD7:

- Si se cumple una o más de las siguientes condiciones, ambas componentes de mvLXCol se establecen igual a 0 y availableFlagLXCol se establece igual a 0.

- colPu se codifica en un modo de intrapredicción.

- colPu se marca como "no disponible".

- pic\_temporal\_mvp\_enable\_flag es igual a 0.

- De lo contrario, el vector de movimiento mvCol, el índice de referencia reldxCol y el identificador de lista de referencia listCol se obtienen como sigue.

- Si PredFlagL0[xPCol][yPCol] es igual a 0, mvCol, reldxCol y listCol se establecen igual a MvL1[xPCol][yPCol], RefldxL1[xPCol][yPCol] y L1, respectivamente.

- De lo contrario (PredFlagL0[xPCol][yPCol] es igual a 1), se aplica lo siguiente.

- Si PredFlagL1[xPCol][yPCol] es igual a 0, mvCol, reldxCol y listCol se establecen igual a MvL0[xPCol][yPCol], RefldxL0[xPCol][yPCol] y L0, respectivamente.

- De lo contrario (PredFlagL1[xPCol][yPCol] es igual a 1), se realizan las siguientes asignaciones.

- Si PicOrderCnt(pic) de cada imagen pic en cada lista de imágenes de referencia es menor que o igual a PicOrderCntVal, mvCol, reldxCol y listCol se establecen igual a MvLX[xPCol][yPCol], RefldxLX[xPCol][yPCol] y LX, respectivamente, siendo X el valor de X para el que se invoca este proceso.

- De lo contrario (PicOrderCnt(pic) de al menos una imagen pic en al menos una lista de imágenes de referencia es mayor que PicOrderCntVal), mvCol, reldxCol y listCol se establecen igual a MvLN[xPCol][yPCol], RefldxLN[xPCol][yPCol] y LN, respectivamente, siendo N el valor de collocated\_from\_10\_flag.

- Si se cumple una de las siguientes condiciones, la variable availableFlagLXCol se establece igual a 0:

- RefPicListX[reldxLX] es una imagen de referencia a largo plazo y LongTermRefPic(colPic, reldxCol, listCol) es igual a 0;

- RefPicListX[reldxLX] es una imagen de referencia a corto plazo y LongTermRefPic(colPic, reldxCol, listCol) es igual a 1;

- De lo contrario, la variable availableFlagLXCol se establece igual a 1 y se aplica lo siguiente.

- Si RefPicListX[reldxLX] es una imagen de referencia a largo plazo, o LongTermRefPic(colPic, reldxCol, listCol) es igual a 1, o PicOrderCnt(colPic) - RefPicOrderCnt(colPic, reldxCol, listCol) es igual a PicOrderCntVal-

PicOrderCnt(RefPicListX[reldxLX])),

mvLXCol = mvCol (8-143)

- De lo contrario, mvLXCol se obtiene como una versión escalada del vector de movimiento mvCol como se especifica a continuación.

$$tx = (16384 + (\text{Abs}(td) \gg 1))/td \quad (8-144)$$

$$\text{DistScaleFactor} = \text{Clip3}(-4096, 4095, (tb * tx + 32) \gg 6) \quad (8-145)$$

$$mvLXCol =$$

$$\begin{aligned} &\text{Clip3}(-8192, 8191.75, \text{Sign}(\text{DistScaleFactor} * mvCol) * \\ &((\text{Abs}(\text{DistScaleFactor} * mvLXCol) + 127) \gg 8)) \end{aligned} \quad (8-146)$$

donde td y tb pueden obtenerse como

$$td = \text{Clip3}(-128, 127, \text{PicOrderCntVal}(\text{colPic}) - \text{RefPicOrderCnt}(\text{colPic}, \text{refIdxCol}, \text{listCol})) \quad (8-147)$$

$$5 \quad td = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListX}[\text{refIdxLX}])) \quad (8-148)$$

En el ejemplo anterior, la disponibilidad del bloque coubicado usado durante la TMVP también puede depender del tipo de imagen (por ejemplo, si la imagen es una imagen de referencia a largo plazo o a corto plazo) de una imagen de referencia para el bloque coubicado. Es decir, incluso cuando hay disponible un bloque inferior derecho para TMVP (después de la etapa 1 en la subcláusula), el bloque inferior derecho se puede establecer además para que no esté disponible si el vector de movimiento en el bloque hace referencia a un tipo de imagen (corto plazo o largo plazo) que es diferente al de la imagen de referencia objetivo. Del mismo modo, un bloque central se puede usar además para TMVP.

15 Por ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para obtener un predictor de vector de movimiento para un vector de movimiento de luma de acuerdo con el siguiente ejemplo detallado. Los datos de entrada para el proceso, implementado por el codificador de vídeo 20 y el decodificador de vídeo 30, pueden incluir:

- 20 - una ubicación de luma (xP, yP) que especifica la muestra de luma superior izquierda de la unidad de predicción actual en relación con la muestra superior izquierda de la imagen actual,
- variables que especifican la anchura y la altura de la unidad de predicción para luma, nPSW y nPSH,
- 25 - el índice de referencia de la partición de unidad de predicción actual refIdxLX (donde X es 0 o 1).

Los datos de salida del proceso pueden incluir:

- 30 - la predicción de vectores de movimiento mvLXCol,
- la bandera de disponibilidad availableFlagLXCol.

La función RefPicOrderCnt(picX, refIdx, LX), cuando se ejecuta mediante el codificador de vídeo 20 y/o el decodificador de vídeo 30, puede devolver el recuento de orden de imagen PicOrderCntVal de la imagen de referencia con índice refIdx de la lista de imágenes de referencia LX de la imagen picX. Un ejemplo de implementación de esta función se especifica como sigue:

$$\text{RefPicOrderCnt}(\text{picX}, \text{refIdx}, \text{LX}) = \text{PicOrderCnt}(\text{RefPicListX}[\text{refIdx}] \text{ de la imagen picX}) \quad (8-141)$$

40 Dependiendo de los valores de slice\_type, collocated\_from\_10\_flag y collocated\_ref\_idx, la variable colPic, que especifica la imagen que contiene la partición coubicada, puede obtenerse como sigue:

- Si el tipo de segmento es igual a B y collocated\_from\_10\_flag es igual a 0, la variable colPic especifica la imagen que contiene la partición coubicada según lo especificado por RefPicList1[collocated\_ref\_idx].
- 45 - De lo contrario (slice\_type es igual a B y collocated\_from\_10\_flag es igual a 1 o slice\_type es igual a P), la variable colPic especifica la imagen que contiene la partición coubicada según lo especificado por RefPicList0[collocated\_ref\_idx].

50 El codificador de vídeo 20 y el decodificador de vídeo 30 pueden obtener la variable colPu y su posición (xPCol, yPCol) usando las siguientes etapas ordenadas:

1. El codificador de vídeo 20 y el decodificador de vídeo 30 pueden obtener la variable colPu como sigue:

$$yPRb = yP + nPSH \quad (8-139)$$



- Si  $(yP \gg \text{Log2CtbSize})$  es igual a  $(yPRb \gg \text{Log2CtbSize})$ , la componente horizontal de la posición de luma inferior derecha de la unidad de predicción actual puede definirse mediante

$$xPRb = xP + nPSW \quad (8-140)$$

5

y la variable colPu puede establecerse como la unidad de predicción que cubre la posición modificada dada por  $((xPRb \gg 4) \ll 4, (yPRb \gg 4) \ll 4)$  dentro de colPic.

10 - De lo contrario  $((yP \gg \text{Log2CtbSize})$  no es igual a  $(yPRb \gg \text{Log2CtbSize}))$ , el codificador de vídeo 20 y el descodificador de vídeo 30 pueden marcar colPu como "no disponible".

2. Cuando colPu se codifica en un modo de intrapredicción o colPu se marca como "no disponible", se aplica lo siguiente, en este ejemplo:

15

- La posición de luma central de la unidad de predicción actual se define mediante

$$xPCtr = (xP + nPSW \gg 1) \quad (8-141)$$

$$yPCtr = (yP + nPSH \gg 1) \quad (8-142)$$

20 - La variable colPu se establece como la unidad de predicción que cubre la posición modificada dada por  $((xPCtr \gg 4) \ll 4, (yPCtr \gg 4) \ll 4)$  dentro de colPic.

3. El codificador de vídeo 20 y el descodificador de vídeo 30 pueden establecer  $(xPCol, yPCol)$  igual a la muestra de luma superior izquierda de colPu con respecto a la muestra de luma superior izquierda de colPic.

25

La función LongTermRefPic(picX, reldx, LX) se puede definir como sigue: Si la imagen de referencia con índice reldx de la lista de imágenes de referencia LX de la imagen picX se marcó como "usada como referencia a largo plazo" en el momento en que picX era la imagen actual, LongTermRefPic(picX, reldx, LX) devuelve 1; de lo contrario, LongTermRefPic(picX, reldx, LX) devuelve 0.

30

El codificador de vídeo 20 y el descodificador de vídeo 30 pueden obtener las variables mvLXCol y availableFlagLXCol como sigue: availableFlagLXCol se establece en 0, numTestBlock es igual a 0.

Si bien numTestBlock es menor que 2 y availableFlagLXCol es igual a 0, lo siguiente se realiza en orden.

35

$$xPCtr = (xP + nPSW \gg 1)$$

$$yPCtr = (yP + nPSH \gg 1)$$

- Si colPu cubre la posición dada por  $((xPCtr \gg 4) \ll 4, (yPCtr \gg 4) \ll 4)$  dentro de colPic, numTestBlock se establece en 1;

40

- De lo contrario, si numTestBlock es igual a 1, colPu se establece como la unidad de predicción que cubre la posición modificada dada por  $((xPCtr \gg 4) \ll 4, (yPCtr \gg 4) \ll 4)$  dentro de colPic, y  $(xPCol, yPCol)$  se establece igual a la muestra de luma superior izquierda de colPu con respecto a la muestra de luma superior izquierda de colPic.

45 - numTestBlock ++

- Si se cumple una o más de las siguientes condiciones, ambas componentes de mvLXCol se establecen igual a 0 y availableFlagLXCol se establece igual a 0.

50 - colPu se codifica en un modo de intrapredicción.

- colPu se marca como "no disponible".

- pic\_temporal\_mvp\_enable\_flag es igual a 0.

55

- De lo contrario, el vector de movimiento mvCol, el índice de referencia reldxCol y el identificador de lista de referencia listCol se obtienen como sigue.

- Si PredFlagL0[xPCol][yPCol] es igual a 0, mvCol, reldxCol y listCol se establecen igual a MvL1[xPCol][yPCol], RefldxL1[xPCol][yPCol] y L1, respectivamente.

60

- De lo contrario (PredFlagL0[xPCol][yPCol] es igual a 1), se aplica lo siguiente.

- Si PredFlagL1[xPCol][yPCol] es igual a 0, mvCol, refIdxCol y listCol se establecen igual a MvL0[xPCol][yPCol], RefIdxL0[xPCol][yPCol] y L0, respectivamente.

- De lo contrario (PredFlagL1[xPCol][yPCol] es igual a 1), se realizan las siguientes asignaciones.

- Si PicOrderCnt(pic) de cada imagen pic en cada lista de imágenes de referencia es menor que o igual a PicOrderCntVal, mvCol, refIdxCol y listCol se establecen igual a MvLX[xPCol][yPCol], RefIdxLX[xPCol][yPCol] y LX, respectivamente, siendo X el valor de X para el que se invoca este proceso.

- De lo contrario (PicOrderCnt(pic) de al menos una imagen pic en al menos una lista de imágenes de referencia es mayor que PicOrderCntVal), mvCol, refIdxCol y listCol se establecen igual a MvLN[xPCol][yPCol], RefIdxLN[xPCol][yPCol] y LN, respectivamente, siendo N el valor de collocated\_from\_10\_flag.

- Si se cumple una de las siguientes condiciones, la variable availableFlagLXCol se establece igual a 0:

- RefPicListX[refIdxLX] es una imagen de referencia a largo plazo y LongTermRefPic(colPic, refIdxCol, listCol) es igual a 0;

- RefPicListX[refIdxLX] es una imagen de referencia a corto plazo y LongTermRefPic(colPic, refIdxCol, listCol) es igual a 1;

- De lo contrario, la variable availableFlagLXCol se establece igual a 1 y se aplica lo siguiente.

- Si RefPicListX[refIdxLX] es una imagen de referencia a largo plazo, o LongTermRefPic(colPic, refIdxCol, listCol) es igual a 1, o PicOrderCnt(colPic) - RefPicOrderCnt(colPic, refIdxCol, listCol) es igual a PicOrderCntVal - PicOrderCnt(RefPicListX[refIdxLX]),

$$mvLXCol = mvCol \cdot (8-143)$$

- De lo contrario, mvLXCol se obtiene como una versión escalada del vector de movimiento mvCol como se especifica a continuación.

$$tx = (16384 + (Abs(td) \gg 1)) / td \quad (8-144)$$

$$DistScaleFactor = Clip3(-4096, 4095, (tb * tx + 32 \gg 6)) \quad (8-145)$$

$$mvLXCol = Clip3(-8192, 8191.75, Sign(DistScaleFactor * mvCol) * ((Abs(DistScaleFactor * mvCol) + 127) \gg 8)) \quad (8-146)$$

donde td y tb pueden obtenerse como:

$$td = Clip3(-128, 127, PicOrderCntVal(colPic) - RefPicOrderCnt(colPic, refIdxCol, listCol)) \quad (8-147)$$

$$tb = Clip3(-128, 127, PicOrderCntVal - PicOrderCnt(RefPicListX[refIdxLX])) \quad (8-148)$$

En un ejemplo alternativo, un vector de movimiento a largo plazo nunca se predice a partir de otro vector de movimiento a largo plazo si los valores de POC de las imágenes de referencia no son los mismos. El codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse de acuerdo con el siguiente proceso para obtener candidatos a predictor de vector de movimiento, donde el texto subrayado representa cambios con respecto a HEVC WD7.

La variable isScaledFlagLX, donde X es 0 o 1, puede establecerse igual a 0. El vector de movimiento mvLXA y la bandera de disponibilidad availableFlagLXA se obtienen en las siguientes etapas ordenadas, donde las elipsis representan texto que es idéntico al del borrador actual de HEVC y los subrayados representan cambios con respecto al borrador actual de HEVC:

1. ...

2. ...

3. ...

4. ...

5. Cuando availableFlagLXA es igual a 0, para  $(x_{A_k}, y_{A_k})$  desde  $(x_{A_0}, y_{A_0})$  hasta  $(x_{A_1}, y_{A_1})$  donde  $y_{A_1} = y_{A_0} - \text{MinPuSize}$ , lo siguiente se aplica repetidamente hasta que availableFlagLXA sea igual a 1:

- Si la unidad de predicción que cubre la ubicación de luma  $(x_{A_k}, y_{A_k})$  está disponible [Ed. (BB): se reescribe usando MinCbAddrZS[ ][ ] y el proceso de disponibilidad para bloques de codificación mínimos], PredMode no es MODE\_INTRA, predFlagLX[ $x_{A_k}$ ][ $y_{A_k}$ ] es igual a 1 y RefPicListX[refldxLX] y RefPicListX[refldxLX[ $x_{A_k}$ ][ $y_{A_k}$ ]] son ambas imágenes de referencia a largo plazo con diferentes valores de POC o son ambas imágenes de referencia a corto plazo, availableFlagLXA se establece igual a 1, el vector de movimiento mvLXA se establece igual al vector de movimiento mvLX[ $x_{A_k}$ ][ $y_{A_k}$ ], refldxA se establece igual a refldxLX[ $x_{A_k}$ ][ $y_{A_k}$ ], ListA se establece igual a ListX.

- De lo contrario, si la unidad de predicción que cubre la ubicación de luma  $(x_{A_k}, y_{A_k})$  está disponible [Ed. (BB): se reescribe usando MinCbAddrZS[ ][ ] y el proceso de disponibilidad para bloques de codificación mínimos], PredMode no es MODE\_INTRA, predFlagLY[ $x_{A_k}$ ][ $y_{A_k}$ ] (con  $Y = !X$ ) es igual a 1, y RefPicListX[refldxLX] y RefPicListY[refldxLY[ $x_{A_k}$ ][ $y_{A_k}$ ]] son ambas imágenes de referencia a largo plazo con diferentes valores de POC o son ambas imágenes de referencia a corto plazo, availableFlagLXA se establece igual a 1, el vector de movimiento mvLXA se establece igual al vector de movimiento mvLY[ $x_{A_k}$ ][ $y_{A_k}$ ], refldxA se establece igual a refldxLY[ $x_{A_k}$ ][ $y_{A_k}$ ], ListA se establece igual a ListY.

- Cuando availableFlagLXA es igual a 1, y tanto RefPicListA[refldxA] como RefPicListX[refldxLX] son imágenes de referencia a corto plazo, mvLXA se obtiene como se especifica a continuación.

$$Tx = (16384 + (\text{Abs}(td) \gg 1)) / td \quad (8-126)$$

$$\text{DistScaleFactor} = \text{Clip3}(-4096, 4095, (tb * tx + 32) \gg 6) \quad (8-127)$$

$$\begin{aligned} mvLXA &= \text{Clip3}(-8192, 8191.75, \text{Sign}(\text{DistScaleFactor} * mvLXA) * \\ &((\text{Abs}(\text{DistScaleFactor} * mvLXA) + 127) \gg 8)) \end{aligned} \quad (8-128)$$

donde td y tb pueden obtenerse como

$$\begin{aligned} td &= \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \\ &\text{PicOrderCnt}(\text{RefPicListA}[\text{refldxA}])) \end{aligned} \quad (8-129)$$

$$\begin{aligned} td &= \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \\ &\text{PicOrderCnt}(\text{RefPicListA}[\text{refldxLX}])) \end{aligned} \quad (8-130)$$

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden obtener el vector de movimiento mvLXB y la bandera de disponibilidad availableFlagLXB usando las siguientes etapas ordenadas, donde nuevamente las elipsis representan texto que es idéntico al de HEVC WD7:

1. ...

2. ...

3. ...

4. ...

5. ...

6. Cuando isScaledFlagLX es igual a 0, availableFlagLXB se establece igual a 0 y para  $(x_{B_k}, y_{B_k})$  desde  $(x_{B_0}, y_{B_0})$  hasta  $(x_{B_2}, y_{B_2})$  donde  $x_{B_0} = x_P + \text{nPSW}$ ,  $x_{B_1} = x_{B_0} - \text{MinPuSize}$ , y  $x_{B_2} = x_P - \text{MinPuSize}$ , lo siguiente se aplica repetidamente hasta que availableFlagLXB sea igual a 1:

- Si la unidad de predicción que cubre la ubicación de luma  $(x_{B_k}, y_{B_k})$  está disponible [Ed. (BB): se reescribe usando MinCbAddrZS[ ][ ] y el proceso de disponibilidad para bloques de codificación mínimos], PredMode no es MODE\_INTRA, predFlagLX[ $x_{B_k}$ ][ $y_{B_k}$ ] es igual a 1 y RefPicListX[refldxLX] y RefPicListX[refldxLX[ $x_{B_k}$ ][ $y_{B_k}$ ]] son ambas imágenes de referencia a largo plazo con diferentes valores de POC o son ambas imágenes de referencia a corto plazo, availableFlagLXB se establece igual a 1, el vector de movimiento mvLXB se establece igual al vector de movimiento mvLX[ $x_{B_k}$ ][ $y_{B_k}$ ], refldxB se establece igual a refldxLX[ $x_{B_k}$ ][ $y_{B_k}$ ], ListB se establece igual a ListX.

- De lo contrario, si la unidad de predicción que cubre la ubicación de luma  $(x_{B_k}, y_{B_k})$  está disponible [Ed. (BB): se reescribe usando MinCbAddrZS[ ][ ] y el proceso de disponibilidad para bloques de codificación mínimos], PredMode

no es MODE\_INTRA,  $\text{predFlagLY}[x_{B_k}][y_{B_k}]$  (con  $Y = !X$ ) es igual a 1, y  $\text{RefPicListX}[\text{refldxLX}]$  y  $\text{RefPicListY}[\text{refldxLY}[x_{B_k}][y_{B_k}]]$  son ambas imágenes de referencia a largo plazo con diferentes valores de POC o son ambas imágenes de referencia a corto plazo,  $\text{availableFlagLXB}$  se establece igual a 1, el vector de movimiento  $\text{mvLXB}$  se establece igual al vector de movimiento  $\text{mvLY}[x_{B_k}][y_{B_k}]$ ,  $\text{refldxB}$  se establece igual a  $\text{refldxLY}[x_{B_k}][y_{B_k}]$ ,  $\text{ListB}$  se establece igual a  $\text{ListY}$ .

- Cuando  $\text{availableFlagLXB}$  es igual a 1 y  $\text{PicOrderCnt}(\text{RefPicListB}[\text{refldxB}])$  no es igual a  $\text{PicOrderCnt}(\text{RefPicListX}[\text{refldxLX}])$  y tanto  $\text{RefPicListB}[\text{refldxB}]$  como  $\text{RefPicListX}[\text{refldxLX}]$  son imágenes de referencia de corto plazo,  $\text{mvLXB}$  se obtiene como se especifica a continuación.

$$tx = (16384 + (\text{Abs}(td) \gg 1)) / td \quad (8-134)$$

$$\text{DistScaleFactor} = \text{Clip3}(-4096, 4095, (tb * tx) \gg 6) \quad (8-135)$$

$$\text{mvLXB} = \text{Clip3}(-8192, 8191.75, \text{Sign}(\text{DistScaleFactor} * \text{mvLXA}) * ((\text{Abs}(\text{DistScaleFactor} * \text{mvLXA}) + 127) \gg 8)) \quad (8-136)$$

donde  $td$  y  $tb$  pueden obtenerse como

$$td = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListB}[\text{refldxB}])) \quad (8-137)$$

$$td = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListX}[\text{refldxLX}])) \quad (8-138)$$

El codificador de vídeo 20 y el descodificador de vídeo 30 pueden configurarse para obtener predictores de vector de movimiento de luma temporal de acuerdo con el siguiente proceso, donde el texto subrayado representa cambios con respecto a HEVC WD7. Las variables  $\text{mvLXCol}$  y  $\text{availableFlagLXCol}$  pueden obtenerse como sigue:

- Si se cumple una o más de las siguientes condiciones, ambas componentes de  $\text{mvLXCol}$  se establecen igual a 0 y  $\text{availableFlagLXCol}$  se establece igual a 0.

-  $\text{colPu}$  se codifica en un modo de intrapredicción.

-  $\text{colPu}$  se marca como "no disponible".

-  $\text{pic\_temporal\_mvp\_enable\_flag}$  es igual a 0.

- De lo contrario, el vector de movimiento  $\text{mvCol}$ , el índice de referencia  $\text{refldxCol}$  y el identificador de lista de referencia  $\text{listCol}$  se obtienen como sigue.

- Si  $\text{PredFlagL0}[x_{PCol}][y_{PCol}]$  es igual a 0,  $\text{mvCol}$ ,  $\text{refldxCol}$  y  $\text{listCol}$  se establecen igual a  $\text{MvL1}[x_{PCol}][y_{PCol}]$ ,  $\text{RefldxL1}[x_{PCol}][y_{PCol}]$  y L1, respectivamente.

- De lo contrario ( $\text{PredFlagL0}[x_{PCol}][y_{PCol}]$  es igual a 1), se aplica lo siguiente.

- Si  $\text{PredFlagL1}[x_{PCol}][y_{PCol}]$  es igual a 0,  $\text{mvCol}$ ,  $\text{refldxCol}$  y  $\text{listCol}$  se establecen igual a  $\text{MvL0}[x_{PCol}][y_{PCol}]$ ,  $\text{RefldxL0}[x_{PCol}][y_{PCol}]$  y L0, respectivamente.

- De lo contrario ( $\text{PredFlagL1}[x_{PCol}][y_{PCol}]$  es igual a 1), se realizan las siguientes asignaciones.

- Si  $\text{PicOrderCnt}(\text{pic})$  de cada imagen  $\text{pic}$  en cada lista de imágenes de referencia es menor que o igual a  $\text{PicOrderCntVal}$ ,  $\text{mvCol}$ ,  $\text{refldxCol}$  y  $\text{listCol}$  se establecen igual a  $\text{MvLX}[x_{PCol}][y_{PCol}]$ ,  $\text{RefldxLX}[x_{PCol}][y_{PCol}]$  y LX, respectivamente, siendo X el valor de X para el que se invoca este proceso.

- De lo contrario ( $\text{PicOrderCnt}(\text{pic})$  de al menos una imagen  $\text{pic}$  en al menos una lista de imágenes de referencia es mayor que  $\text{PicOrderCntVal}$ ),  $\text{mvCol}$ ,  $\text{refldxCol}$  y  $\text{listCol}$  se establecen igual a  $\text{MvLN}[x_{PCol}][y_{PCol}]$ ,  $\text{RefldxLN}[x_{PCol}][y_{PCol}]$  y LN, respectivamente, siendo N el valor de  $\text{collocated\_from\_10\_flag}$ .

- Si se cumple una de las siguientes condiciones, la variable  $\text{availableFlagLXCol}$  se establece igual a 0:

-  $\text{RefPicListX}[\text{refldxLX}]$  es una imagen de referencia a largo plazo y  $\text{LongTermRefPic}(\text{colPic}, \text{refldxCol}, \text{listCol})$  es igual a 0:

- RefPicListX[refldxLX] es una imagen de referencia a corto plazo y LongTermRefPic(colPic, refldxCol, listCol) es igual a 1;

5 - RefPicListX[refldxLX] es una imagen de referencia a largo plazo, LongTermRefPic(colPic, refldxCol, listCol) es igual a 1 y RefPicOrderCnt(colPic, refldxCol, listCol) no es igual a PicOrderCnt(RefPicListX[refldxLX]).

- De lo contrario, la variable `availableFlagLXCol` se establece igual a 1 y se aplica lo siguiente.

10 - Si RefPicListX[refldxLX] es una imagen de referencia a largo plazo, o LongTermRefPic(colPic, refldxCol, listCol) es igual a 1, o PicOrderCnt(colPic) - RefPicOrderCnt(colPic, refldxCol, listCol) es igual a PicOrderCntVal - PicOrderCnt(RefPicListX[refldxLX]),

$$mvLXCol = mvCol \quad (8-143)$$

15 - De lo contrario, `mvLXCol` se obtiene como una versión escalada del vector de movimiento `mvCol` como se especifica a continuación.

$$tx = (16384 + (Abs(td) \gg 1)) / td \quad (8-144)$$

$$DistScaleFactor = Clip3(-4096, 4095, (tb * tx + 32) \gg 6) \quad (8-145)$$

$$mvLXCol =$$

$$Clip3(-8192, 8191.75, Sign(DistScaleFactor * mvCol) * ((Abs(DistScaleFactor * mvCol) + 127) \gg 8)) \quad (8-146)$$

20 donde `td` y `tb` pueden obtenerse como

$$td = Clip3(-128, 127, PicOrderCntVal(colPic) - RefPicOrderCnt(colPic, refldxCol, listCol)) \quad (8-147)$$

$$tb = Clip3(-128, 127, PicOrderCntVal - PicOrderCnt(RefPicListX[refldxLX])) \quad (8-148)$$

25 Como otro ejemplo más, una imagen de referencia entre vistas puede marcarse como "no usada como referencia". Para simplificar, imagen de este tipo puede denominarse imagen que no es de referencia en la especificación base de HEVC, y una imagen marcada como "usada como referencia a largo plazo" o "usada como referencia a corto plazo" puede denominarse imagen de referencia en la especificación base de HEVC. Los términos "imagen de referencia" e "imagen que no es de referencia" pueden reemplazarse por "imagen marcada como "usada como referencia" e "imagen marcada como "no usada como referencia".

30 La función `UnusedRefPic(picX, refldx, LX)` se puede definir como sigue. Si la imagen de referencia con índice `refldx` de la lista de imágenes de referencia `LX` de la imagen `picX` se marcó como "no usada como referencia" en el momento en que `picX` era la imagen actual, `UnusedRefPic(picX, refldx, LX)` devuelve 1; de lo contrario, `UnusedRefPic(picX, refldx, LX)` devuelve 0.

35 El codificador de vídeo 20 y el descodificador de vídeo 30 pueden configurarse para realizar un proceso de obtención para candidatos a predictor de vector de movimiento como sigue, donde el texto subrayado representa cambios con respecto a HEVC WD7 y las elipsis representan texto que es idéntico al de HEVC WD7:

40 La variable `isScaledFlagLX`, donde `X` es 0 o 1, puede establecerse igual a 0.

El vector de movimiento `mvLXA` y la bandera de disponibilidad `availableFlagLXA` pueden obtenerse en las siguientes etapas ordenadas:

45 1. ...

2. ...

3. ...

50 4. ...

5. Cuando `availableFlagLXA` es igual a 0, para  $(xA_k, yA_k)$  desde  $(xA_0, yA_0)$  hasta  $(xA_1, yA_1)$  donde  $yA_1 = yA_0 - \text{MinPulseSize}$ , lo siguiente se aplica repetidamente hasta que `availableFlagLXA` sea igual a 1:

55

- Si la unidad de predicción que cubre la ubicación de luma ( $x_{A_k}$ ,  $y_{A_k}$ ) está disponible [Ed. (BB): se reescribe usando MinCbAddrZS[ ] y el proceso de disponibilidad para bloques de codificación mínimos], PredMode no es MODE\_INTRA, predFlagLX[ $x_{A_k}$ ][ $y_{A_k}$ ] es igual a 1 y RefPicListX[refldxLX] y RefPicListX[refldxLX[ $x_{A_k}$ ][ $y_{A_k}$ ]] son ambas imágenes de referencia o no son ninguna imágenes de referencia, availableFlagLXA se establece igual a 1, el vector de movimiento mvLXA se establece igual al vector de movimiento mvLX[ $x_{A_k}$ ][ $y_{A_k}$ ], refldxA se establece igual a refldxLX[ $x_{A_k}$ ][ $y_{A_k}$ ], ListA se establece igual a ListX.

- De lo contrario, si la unidad de predicción que cubre la ubicación de luma ( $x_{A_k}$ ,  $y_{A_k}$ ) está disponible [Ed. (BB): se reescribe usando MinCbAddrZS[ ] y el proceso de disponibilidad para bloques de codificación mínimos], PredMode no es MODE\_INTRA, predFlagLY[ $x_{A_k}$ ][ $y_{A_k}$ ] (con  $Y = !X$ ) es igual a 1, y RefPicListX[refldxLX] y RefPicListY[refldxLY[ $x_{A_k}$ ][ $y_{A_k}$ ]] son ambas imágenes de referencia o no son ninguna imágenes de referencia, availableFlagLXA se establece igual a 1, el vector de movimiento mvLXA se establece igual al vector de movimiento mvLY[ $x_{A_k}$ ][ $y_{A_k}$ ], refldxA se establece igual a refldxLY[ $x_{A_k}$ ][ $y_{A_k}$ ], ListA se establece igual a ListY.

- Cuando availableFlagLXA es igual a 1, y tanto RefPicListA[refldxA] como RefPicListX[refldxLX] son imágenes de referencia a corto plazo, mvLXA se obtiene como se especifica a continuación.

$$Tx = (16384 + (\text{Abs}(td) \gg 1)) / td \quad (8-126)$$

$$\text{DistScaleFactor} = \text{Clip3}(-4096, 4095, (tb * tx + 32) \gg 6) \quad (8-127)$$

$$\begin{aligned} mvLXA &= \text{Clip3}(-8192, 8191.75, \text{Sign}(\text{DistScaleFactor} * mvLXA) * \\ &((\text{Abs}(\text{DistScaleFactor} * mvLXA) + 127) \gg 8)) \end{aligned} \quad (8-128)$$

donde td y tb pueden obtenerse como

$$td = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListA}[\text{refldxA}])) \quad (8-129)$$

$$tb = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListA}[\text{refldxLX}])) \quad (8-130)$$

El codificador de vídeo 20 y el descodificador de vídeo 30 pueden configurarse para obtener el vector de movimiento mvLXB y la bandera de disponibilidad availableFlagLXB usando las siguientes etapas ordenadas, donde el texto subrayado representa cambios con respecto a HEVC WD7:

1. ...

2. ...

3. ...

4. ...

5. ...

6. Cuando isScaledFlagLX es igual a 0, availableFlagLXB se establece igual a 0 y para ( $x_{B_k}$ ,  $y_{B_k}$ ) desde ( $x_{B_0}$ ,  $y_{B_0}$ ) hasta ( $x_{B_2}$ ,  $y_{B_2}$ ) donde  $x_{B_0} = x_P + nPSW$ ,  $x_{B_1} = x_{B_0} - \text{MinPuSize}$ , y  $x_{B_2} = x_P - \text{MinPuSize}$ , lo siguiente se aplica repetidamente hasta que availableFlagLXB sea igual a 1:

- Si la unidad de predicción que cubre la ubicación de luma ( $x_{B_k}$ ,  $y_{B_k}$ ) está disponible [Ed. (BB): se reescribe usando MinCbAddrZS[ ] y el proceso de disponibilidad para bloques de codificación mínimos], PredMode no es MODE\_INTRA, predFlagLX[ $x_{B_k}$ ][ $y_{B_k}$ ] es igual a 1 y RefPicListX[refldxLX] y RefPicListX[refldxLX[ $x_{B_k}$ ][ $y_{B_k}$ ]] son ambas imágenes de referencia o no son ninguna imágenes de referencia, availableFlagLXB se establece igual a 1, el vector de movimiento mvLXB se establece igual al vector de movimiento mvLX[ $x_{B_k}$ ][ $y_{B_k}$ ], refldxB se establece igual a refldxLX[ $x_{B_k}$ ][ $y_{B_k}$ ], ListB se establece igual a ListX.

- De lo contrario, si la unidad de predicción que cubre la ubicación de luma ( $x_{B_k}$ ,  $y_{B_k}$ ) está disponible [Ed. (BB): se reescribe usando MinCbAddrZS[ ] y el proceso de disponibilidad para bloques de codificación mínimos], PredMode no es MODE\_INTRA, predFlagLY[ $x_{B_k}$ ][ $y_{B_k}$ ] (con  $Y = !X$ ) es igual a 1, y RefPicListX[refldxLX] y RefPicListY[refldxLY[ $x_{B_k}$ ][ $y_{B_k}$ ]] son ambas imágenes de referencia o no son ninguna imágenes de referencia, availableFlagLXB se establece igual a 1, el vector de movimiento mvLXB se establece igual al vector de movimiento mvLY[ $x_{B_k}$ ][ $y_{B_k}$ ], refldxB se establece igual a refldxLY[ $x_{B_k}$ ][ $y_{B_k}$ ], ListB se establece igual a ListY.

- Cuando availableFlagLXB es igual a 1 y PicOrderCnt(RefPicListB[refIdxB]) no es igual a PicOrderCnt(RefPicListX[refIdxLX]) y tanto RefPicListB[refIdxB] como RefPicListX[refIdxLX] son imágenes de referencia de corto plazo, mvLXB se obtiene como se especifica a continuación.

$$tx = (16384 + (\text{Abs}(td) \gg 1)) / td \quad (8-134)$$

$$\text{DistScaleFactor} = \text{Clip3}(-4096, 4095, (tb * tx + 32) \gg 6) \quad (8-135)$$

$$\begin{aligned} mvLXB &= \text{Clip3}(-8192, 8191.75, \text{Sign}(\text{DistScaleFactor} * mvLXA) * \\ &((\text{Abs}(\text{DistScaleFactor} * mvLXA) + 127) \gg 8)) \quad (8-136) \end{aligned}$$

5

- donde td y tb pueden obtenerse como

$$td = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListB}[\text{refIdxB}])) \quad (8-137)$$

$$td = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListX}[\text{refIdxLX}])) \quad (8-138)$$

10

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para obtener predictores de vector de movimiento de luma temporal como sigue, donde el texto subrayado representa cambios con respecto a HEVC WD7:

Las variables mvLXCol y availableFlagLXCol pueden obtenerse como sigue.

15

- Si se cumple una o más de las siguientes condiciones, ambas componentes de mvLXCol se establecen igual a 0 y availableFlagLXCol se establece igual a 0.

- colPu se codifica en un modo de intrapredicción.

20

- colPu se marca como "no disponible".

- pic\_temporal\_mvp\_enable\_flag es igual a 0.

25

- De lo contrario, el vector de movimiento mvCol, el índice de referencia refIdxCol y el identificador de lista de referencia listCol se obtienen como sigue.

- Si PredFlagL0[xPCol][yPCol] es igual a 0, mvCol, refIdxCol y listCol se establecen igual a MvL1[xPCol][yPCol], RefIdxL1[xPCol][yPCol] y L1, respectivamente.

30

- De lo contrario (PredFlagL0[xPCol][yPCol] es igual a 1), se aplica lo siguiente.

- Si PredFlagL1[xPCol][yPCol] es igual a 0, mvCol, refIdxCol y listCol se establecen igual a MvL0[xPCol][yPCol], RefIdxL0[xPCol][yPCol] y L0, respectivamente.

35

- De lo contrario (PredFlagL1[xPCol][yPCol] es igual a 1), se realizan las siguientes asignaciones.

- Si PicOrderCnt(pic) de cada imagen pic en cada lista de imágenes de referencia es menor que o igual a PicOrderCntVal, mvCol, refIdxCol y listCol se establecen igual a MvLX[xPCol][yPCol], RefIdxLX[xPCol][yPCol] y LX, respectivamente, siendo X el valor de X para el que se invoca este proceso.

40

- De lo contrario (PicOrderCnt(pic) de al menos una imagen pic en al menos una lista de imágenes de referencia es mayor que PicOrderCntVal), mvCol, refIdxCol y listCol se establecen igual a MvLN[xPCol][yPCol], RefIdxLN[xPCol][yPCol] y LN, respectivamente, siendo N el valor de collocated\_from\_10\_flag.

45

- Si se cumple una de las siguientes condiciones, la variable availableFlagLXCol se establece igual a 0:

- RefPicListX[refIdxLX] es una imagen que no es de referencia y UnusedRefPic(colPic, refIdxCol, listCol) es igual a 0;

50

- RefPicListX[refIdxLX] es una imagen de referencia y UnusedRefPic(colPic, refIdxCol, listCol) es igual a 1;

- De lo contrario, la variable availableFlagLXCol se establece igual a 1 y se aplica lo siguiente.

- Si RefPicListX[refIdxLX] es una imagen de referencia a largo plazo, o LongTermRefPic(colPic, refIdxCol, listCol) es igual a 1, o PicOrderCnt(colPic) - RefPicOrderCnt(colPic, refIdxCol, listCol) es igual a PicOrderCntVal - PicOrderCnt(RefPicListX[refIdxLX]),

55

$$mvLXCol = mvCol \text{ (8-143)}$$

- De lo contrario, mvLXCol se obtiene como una versión escalada del vector de movimiento mvCol como se especifica a continuación.

$$tx = (16384 + (Abs(td) \gg 1)) / td \text{ (8-144)}$$

$$DistScaleFactor = Clip3(-4096, 4095, (tb * tx + 32) \gg 6) \text{ (8-145)}$$

$$mvLXCol =$$

$$Clip3(-8192, 8191.75, Sign(DistScaleFactor * mvCol) * ((Abs(DistScaleFactor * mvCol) + 127) \gg 8)) \text{ (8-146)}$$

- donde td y tb se obtienen como

$$td = Clip3(-128, 127, PicOrderCntVal(colPic) - RefPicOrderCnt(colPic, reldxCol, listCol)) \text{ (8-147)}$$

$$tb = Clip3(-128, 127, PicOrderCntVal - PicOrderCnt(RefPicListX[reldxLX])) \text{ (8-148)}$$

Con respecto al ejemplo mencionado anteriormente como el "quinto ejemplo", el codificador de vídeo 20 y el descodificador de vídeo 30 pueden configurarse para funcionar de acuerdo con cualquiera o todas las técnicas siguientes. En este ejemplo, la predicción entre vectores de movimiento que hacen referencia a diferentes imágenes de referencia a largo plazo puede estar inhabilitada, la predicción entre vectores de movimiento que hacen referencia a diferentes imágenes de referencia entre vistas puede estar inhabilitada y la predicción entre vectores de movimiento que hacen referencia a una imagen de referencia entre vistas y a una imagen de referencia a largo plazo puede estar inhabilitada.

En este ejemplo, la función AddPicId(pic) devuelve el índice de orden de vista o el ID de capa de la vista o capa a la que pertenece la imagen. Por lo tanto, para cualquier imagen "pic" que pertenezca a la vista o capa base, AddPicId(pic) devuelve 0. En la especificación base de HEVC, se puede aplicar lo siguiente: la función AddPicId(pic) se puede definir como sigue: AddPicId(pic) devuelve 0 (o reserved\_one\_5bits menos 1). En este ejemplo, cuando AddPicId(pic) no es igual a 0, la imagen pic no es una imagen de referencia temporal (es decir, ni una imagen de referencia a corto plazo ni una imagen de referencia a largo plazo). AddPicId(picX, reldx, LX) puede devolver AddPicId(pic), donde pic es la imagen de referencia con índice reldx de la lista de imágenes de referencia LX de la imagen picX.

El codificador de vídeo 20 y el descodificador de vídeo 30 pueden configurarse para realizar un proceso de obtención para candidatos a predictor de vectores de movimiento. Los datos de entrada para este proceso pueden incluir una ubicación de luma (xP, yP) que especifica la muestra de luma superior izquierda de la unidad de predicción actual en relación con la muestra superior izquierda de la imagen actual, variables que especifican la anchura y la altura de la unidad de predicción para luma, nPSW y nPSH, y el índice de referencia de la partición de unidad de predicción actual reldxLX (siendo X 0 o 1). Los datos de salida de este proceso pueden incluir (donde N puede reemplazarse por A o B): los vectores de movimiento mvLXN de las unidades de predicción vecinas y las banderas de disponibilidad availableFlagLXN de las unidades de predicción vecinas.

La variable isScaledFlagLX, donde X es 0 o 1, puede establecerse igual a 0. El codificador de vídeo 20 y el descodificador de vídeo 30 pueden configurarse para obtener el vector de movimiento mvLXA y la bandera de disponibilidad availableFlagLXA usando las siguientes etapas ordenadas, donde el texto subrayado representa cambios con respecto a HEVC WD7:

1. Sea un conjunto de dos ubicaciones de muestra ( $xA_k, yA_k$ ), con  $k = 0, 1$ , que especifica ubicaciones de muestra con  $xA_k = xP - 1$ ,  $yA_0 = yP + nPSH$  e  $yA_1 = yA_0 - MinPuSize$ . El conjunto de ubicaciones de muestra ( $xA_k, yA_k$ ) representa las ubicaciones de muestra justo en el lado izquierdo del límite de partición izquierdo y su línea extendida.

2. Establézcase la bandera de disponibilidad availableFlagLXA inicialmente igual 0 y ambas componentes de mvLXA igual a 0.

3. Cuando una o más de las siguientes condiciones son verdaderas, la variable isScaledFlagLX se establece igual a 1.

- La unidad de predicción que cubre la ubicación de luma ( $xA_0, yA_0$ ) está disponible [Ed. (BB): se reescribe usando MinCbAddrZS[ ][ ] y el proceso de disponibilidad para bloques de codificación mínimos] y PredMode no es MODE\_INTRA.



- La unidad de predicción que cubre la ubicación de luma ( $x_{A_1}$ ,  $y_{A_1}$ ) está disponible [Ed. (BB): se reescribe usando MinCbAddrZS[ ][ ] y el proceso de disponibilidad para bloques de codificación mínimos] y PredMode no es MODE\_INTRA.

4. Para ( $x_{A_k}$ ,  $y_{A_k}$ ) de ( $x_{A_0}$ ,  $y_{A_0}$ ) a ( $x_{A_1}$ ,  $y_{A_1}$ ) donde  $y_{A_1} = y_{A_0} - \text{MinPuSize}$ , si availableFlagLXA es igual a 0, se aplica lo siguiente:

- Si la unidad de predicción que cubre la ubicación de luma ( $x_{A_k}$ ,  $y_{A_k}$ ) está disponible [Ed. (BB): se reescribe usando MinCbAddrZS[ ][ ] y el proceso de disponibilidad para bloques de codificación mínimos], PredMode no es MODE\_INTRA, predFlagLX[ $x_{A_k}$ ][ $y_{A_k}$ ] es igual a 1 y el índice de referencia refIdxLX[ $x_{A_k}$ ][ $y_{A_k}$ ] es igual al índice de referencia de la unidad de predicción actual refIdxLX, availableFlagLXA se establece igual a 1 y el vector de movimiento mvLXA se establece igual al vector de movimiento mvLX[ $x_{A_k}$ ][ $y_{A_k}$ ], refIdxA se establece igual a refIdxLX[ $x_{A_k}$ ][ $y_{A_k}$ ] y ListA se establece igual a ListX.

- De lo contrario, si la unidad de predicción que cubre la ubicación de luma ( $x_{A_k}$ ,  $y_{A_k}$ ) está disponible [Ed. (BB): se reescribe usando MinCbAddrZS[ ][ ] y el proceso de disponibilidad para bloques de codificación mínimos], PredMode no es MODE\_INTRA, predFlagLY[ $x_{A_k}$ ][ $y_{A_k}$ ] (con  $Y \neq X$ ) es igual a 1, AddPicId(RefPicListX[refIdxLX]) es igual a AddPicId(RefPicListY[refIdxLY[ $x_{A_k}$ ][ $y_{A_k}$ ]]), y PicOrderCnt(RefPicListY[refIdxLY[ $x_{A_k}$ ][ $y_{A_k}$ ]]) es igual a PicOrderCnt(RefPicListX[refIdxLX]), availableFlagLXA se establece igual a 1, el vector de movimiento mvLXA se establece igual al vector de movimiento mvLY[ $x_{A_k}$ ][ $y_{A_k}$ ], refIdxA se establece igual a refIdxLY[ $x_{A_k}$ ][ $y_{A_k}$ ], ListA se establece igual a ListY y mvLXA se establece igual a mvLY.

- Cuando availableFlagLXA es igual a 1, availableFlagLXA se establece en 0 si se cumple uno o más de lo siguiente:

- Uno y solo uno de RefPicListX[refIdxLX] y ListA[refIdxA] es una imagen de referencia a largo plazo;

- Tanto RefPicListX[refIdxLX] como ListA[refIdxA] son imágenes de referencia a largo plazo y PicOrderCnt(ListA[refIdxA]) no es igual a PicOrderCnt(RefPicListX[refIdxLX]).

5. Cuando availableFlagLXA es igual a 0, para ( $x_{A_k}$ ,  $y_{A_k}$ ) de ( $x_{A_0}$ ,  $y_{A_0}$ ) a ( $x_{A_1}$ ,  $y_{A_1}$ ) donde  $y_{A_1} = y_{A_0} - \text{MinPuSize}$ , si availableFlagLXA es igual a 0, se aplica lo siguiente:

- Si la unidad de predicción que cubre la ubicación de luma ( $x_{A_k}$ ,  $y_{A_k}$ ) está disponible [Ed. (BB): se reescribe usando MinCbAddrZS[ ][ ] y el proceso de disponibilidad para bloques de codificación mínimos], PredMode no es MODE\_INTRA, predFlagLX[ $x_{A_k}$ ][ $y_{A_k}$ ] es igual a 1, y AddPicId(RefPicListLX[refIdxLX]) es igual a AddPicId(RefPicListLX[refIdxLX[ $x_{A_k}$ ][ $y_{A_k}$ ]]), availableFlagLXA se establece igual a 1, el vector de movimiento mvLXA se establece igual al vector de movimiento mvLX[ $x_{A_k}$ ][ $y_{A_k}$ ], refIdxA se establece igual a refIdxLX[ $x_{A_k}$ ][ $y_{A_k}$ ], ListA se establece igual a ListX.

- De lo contrario, si la unidad de predicción que cubre la ubicación de luma ( $x_{A_k}$ ,  $y_{A_k}$ ) está disponible [Ed. (BB): se reescribe usando MinCbAddrZS[ ][ ] y el proceso de disponibilidad para bloques de codificación mínimos], PredMode no es MODE\_INTRA, predFlagLY[ $x_{A_k}$ ][ $y_{A_k}$ ] (con  $Y \neq X$ ) es igual a 1, y AddPicId(RefPicListLX[refIdxLX]) es igual a AddPicId(RefPicListLY[refIdxLY[ $x_{A_k}$ ][ $y_{A_k}$ ]]), availableFlagLXA se establece igual a 1, el vector de movimiento mvLXA se establece igual al vector de movimiento mvLY[ $x_{A_k}$ ][ $y_{A_k}$ ], refIdxA se establece igual a refIdxLY[ $x_{A_k}$ ][ $y_{A_k}$ ], ListA se establece igual a ListY.

- Cuando availableFlagLXA es igual a 1, availableFlagLXA se establece en 0 si se cumple uno o más de lo siguiente:

- Uno y solo uno de RefPicListX[refIdxLX] y ListA[refIdxA] es una imagen de referencia a largo plazo;

- Tanto RefPicListX[refIdxLX] como ListA[refIdxA] son imágenes de referencia a largo plazo y PicOrderCnt(ListA[refIdxA]) no es igual a PicOrderCnt(RefPicListX[refIdxLX]).

- Cuando availableFlagLXA es igual a 1, y tanto RefPicListA[refIdxA] como RefPicListX[refIdxLX] son imágenes de referencia a corto plazo, mvLXA se obtiene como se especifica a continuación.

$$tx = (16384 + (\text{Abs}(td) \gg 1)) / td \quad (8-126)$$

$$\text{DistScaleFactor} = \text{Clip3}(-4096, 4095, (tb * tx + 32) \gg 6) \quad (8-127)$$

$$mvLXA = \text{Clip3}(-8192, 8191.75, \text{Sign}(\text{DistScaleFactor} * mvLXA) * ((\text{Abs}(\text{DistScaleFactor} * mvLXA) + 127) \gg 8)) \quad (8-128)$$

- donde td y tb pueden obtenerse como

$$td = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListA}[\text{refldxA}])) \quad (8-129)$$

$$td = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListX}[\text{refldxLX}])) \quad (8-130)$$

El codificador de vídeo 20 y el descodificador de vídeo 30 pueden configurarse para obtener el vector de movimiento mvLXB y la bandera de disponibilidad availableFlagLXB usando las siguientes etapas ordenadas, donde el texto subrayado representa cambios con respecto a HEVC WD7:

1. Sea un conjunto de tres ubicaciones de muestra  $(x_{B_k}, y_{B_k})$ , con  $k = 0, 1, 2$ , que especifican ubicaciones de muestra con  $x_{B_0} = x_P + nPSW$ ,  $x_{B_1} = x_{B_0} - \text{MinPuSize}$ ,  $x_{B_2} = x_P - \text{MinPuSize}$  e  $y_{B_k} = y_P - 1$ . El conjunto de ubicaciones de muestra  $(x_{B_k}, y_{B_k})$  representa las ubicaciones de muestra justo en el lado superior del límite de partición superior y su línea extendida. [Ed. (BB): se define MinPuSize en el SPS, pero la obtención debe depender del uso de una bandera AMP]

2. Cuando  $y_P - 1$  es menor que  $((y_C \gg \text{Log2CtbSize}) \ll \text{Log2CtbSize})$ , se aplica lo siguiente.

$$x_{B_0} = (x_{B_0} \gg 3) \ll 3 + ((x_{B_0} \gg 3) \& 1) * 7 \quad (8-131)$$

$$x_{B_1} = (x_{B_1} \gg 3) \ll 3 + ((x_{B_1} \gg 3) \& 1) * 7 \quad (8-132)$$

$$x_{B_2} = (x_{B_2} \gg 3) \ll 3 + ((x_{B_2} \gg 3) \& 1) * 7 \quad (8-133)$$

3. Establézcase la bandera de disponibilidad availableFlagLXB inicialmente igual 0 y ambas componentes de mvLXB igual a 0.

4. Para  $(x_{B_k}, y_{B_k})$  desde  $(x_{B_0}, y_{B_0})$  hasta  $(x_{B_2}, y_{B_2})$ , donde  $x_{B_0} = x_P + nPSW$ ,  $x_{B_1} = x_{B_0} - \text{MinPuSize}$  y  $x_{B_2} = x_P - \text{MinPuSize}$ , si availableFlagLXB es igual a 0, se aplica lo siguiente:

- Si la unidad de predicción que cubre la ubicación de luma  $(x_{B_k}, y_{B_k})$  está disponible [Ed. (BB): se reescribe usando MinCbAddrZS[ ][ ] y el proceso de disponibilidad para bloques de codificación mínimos], PredMode no es MODE\_INTRA, predFlagLX[ $x_{B_k}$ ][ $y_{B_k}$ ] es igual a 1 y el índice de referencia refldxLX[ $x_{B_k}$ ][ $y_{B_k}$ ] es igual al índice de referencia de la unidad de predicción actual refldxLX, availableFlagLXB se establece igual a 1 y el vector de movimiento mvLXB se establece igual al vector de movimiento mvLX[ $x_{B_k}$ ][ $y_{B_k}$ ], refldxB se establece igual a refldxLX[ $x_{B_k}$ ][ $y_{B_k}$ ] y ListB se establece igual a ListX.

- De lo contrario, si la unidad de predicción que cubre la ubicación de luma  $(x_{B_k}, y_{B_k})$  está disponible [Ed. (BB): se reescribe usando MinCbAddrZS[ ][ ] y el proceso de disponibilidad para bloques de codificación mínimos], PredMode no es MODE\_INTRA, predFlagLY[ $x_{B_k}$ ][ $y_{B_k}$ ] (con  $Y = \text{!}X$ ) es igual a 1, AddPicId(RefPicListX[refldxLX]) es igual a AddPicId(RefPicListLY[refldxLY[ $x_{B_k}$ ][ $y_{B_k}$ ]]), y PicOrderCnt(RefPicListY[refldxLY[ $x_{B_k}$ ][ $y_{B_k}$ ]]) es igual a PicOrderCnt(RefPicListX[refldxLX]), availableFlagLXB se establece igual a 1, el vector de movimiento mvLXB se establece igual al vector de movimiento mvLY[ $x_{B_k}$ ][ $y_{B_k}$ ], refldxB se establece igual a refldxLY[ $x_{B_k}$ ][ $y_{B_k}$ ], y ListB se establece igual a ListY.

- Cuando availableFlagLXA es igual a 1, availableFlagLXA se establece en 0 si se cumple uno o más de lo siguiente:

- Uno y solo uno de RefPicListX[refldxLX] y ListB[refldxB] es una imagen de referencia a largo plazo.

- AddPicId(RefPicListX[refldxLX]) no es igual a AddPicId(ListB[refldxB]).

- Tanto RefPicListX[refldxLX] como ListB[refldxB] son imágenes de referencia a largo plazo y PicOrderCnt(ListB[refldxB]) no es igual a PicOrderCnt(RefPicListX[refldxLX]).

5. Cuando isScaledFlagLX es igual a 0 y availableFlagLXB es igual a 1, mvLXA se establece igual a mvLXB y refldxA se establece igual a refldxB y availableFlagLXA se establece igual a 1.

6. Cuando isScaledFlagLX es igual a 0, availableFlagLXB se establece igual a 0 y para  $(x_{B_k}, y_{B_k})$  desde  $(x_{B_0}, y_{B_0})$  hasta  $(x_{B_2}, y_{B_2})$  donde  $x_{B_0} = x_P + nPSW$ ,  $x_{B_1} = x_{B_0} - \text{MinPuSize}$ , y  $x_{B_2} = x_P - \text{MinPuSize}$ , si availableFlagLXB es igual a 0, se aplica lo siguiente:

- Si la unidad de predicción que cubre la ubicación de luma  $(x_{B_k}, y_{B_k})$  está disponible [Ed. (BB): se reescribe usando MinCbAddrZS[ ][ ] y el proceso de disponibilidad para bloques de codificación mínimos], PredMode no es MODE\_INTRA, predFlagLX[ $x_{B_k}$ ][ $y_{B_k}$ ] es igual a 1 y AddPicId(RefPicListX[refldxLX]) es igual a AddPicId(RefPicListX[refldxLX[ $x_{B_k}$ ][ $y_{B_k}$ ]]), availableFlagLXB se establece igual a 1, el vector de movimiento mvLXB

se establece igual al vector de movimiento mvLX[xB<sub>k</sub>][yB<sub>k</sub>], reldxB se establece igual a reldxLX[xB<sub>k</sub>][yB<sub>k</sub>], ListB se establece igual a ListX.

- De lo contrario, si la unidad de predicción que cubre la ubicación de luma (xB<sub>k</sub>, yB<sub>k</sub>) está disponible [Ed. (BB): se reescribe usando MinCbAddrZS[ ] y el proceso de disponibilidad para bloques de codificación mínimos], PredMode no es MODE\_INTRA, predFlagLY[xB<sub>k</sub>][yB<sub>k</sub>] (con Y ≠ X) es igual a 1, y AddPicId(RefPicListX[reldxLX]) es igual a AddPicId(RefPicListY[reldxLY[xB<sub>k</sub>][yB<sub>k</sub>]]), availableFlagLXB se establece igual a 1, el vector de movimiento mvLXB se establece igual al vector de movimiento mvLY[xB<sub>k</sub>][yB<sub>k</sub>], reldxB se establece igual a reldxLY[xB<sub>k</sub>][yB<sub>k</sub>], ListB se establece igual a ListY.

- Cuando availableFlagLXA es igual a 1, availableFlagLXA se establece en 0 si se cumple uno o más de lo siguiente:

- Uno y solo uno de RefPicListX[reldxLX] y ListB[reldxB] es una imagen de referencia a largo plazo.

- Tanto RefPicListX[reldxLX] como ListB[reldxB] son imágenes de referencia a largo plazo y PicOrderCnt(ListB[reldxB]) no es igual a PicOrderCnt(RefPicListX[reldxLX]).

- Cuando availableFlagLXB es igual a 1 y PicOrderCnt(RefPicListB[reldxB]) no es igual a PicOrderCnt(RefPicListX[reldxLX]) y tanto RefPicListB[reldxB] como RefPicListX[reldxLX] son imágenes de referencia de corto plazo, mvLXB se obtiene como se especifica a continuación.

$$tx = (16384 + (\text{Abs}(td) \gg 1)) / td \quad (8-134)$$

$$\text{DistScaleFactor} = \text{Clip3}(-4096, 4095, (tb * tx + 32) \gg 6) \quad (8-135)$$

$$\begin{aligned} mvLXB &= \text{Clip3}(-8192, 8191.75, \text{Sign}(\text{DistScaleFactor} * mvLXA) * \\ &((\text{Abs}(\text{DistScaleFactor} * mvLXA) + 127) \gg 8)) \end{aligned} \quad (8-136)$$

- donde td y tb pueden obtenerse como

$$td = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListB}[\text{reldxB}])) \quad (8-137)$$

$$tb = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListX}[\text{reldxLX}])) \quad (8-138)$$

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para obtener predictores de vector de movimiento de luma temporal como sigue. Los datos de entrada para el proceso pueden incluir una ubicación de luma (xP, yP) que especifica la muestra de luma superior izquierda de la unidad de predicción actual en relación con la muestra superior izquierda de la imagen actual, variables que especifican la anchura y la altura de la unidad de predicción para luma, nPSW y nPSH, y el índice de referencia de la partición de unidad de predicción actual reldxLX (siendo X 0 o 1). Los datos de salida del proceso pueden incluir la predicción de vectores de movimiento mvLXCol y la bandera de disponibilidad availableFlagLXCol.

La función RefPicOrderCnt(picX, reldx, LX), que puede devolver el recuento de orden de imagen PicOrderCntVal de la imagen de referencia con índice reldx de la lista de imágenes de referencia LX de la imagen picX, se puede especificar como sigue:

$$\text{RefPicOrderCnt}(\text{picX}, \text{reldx}, \text{LX}) = \text{PicOrderCnt}(\text{RefPicListX}[\text{reldx}] \text{ de la imagen picX}) \quad (8-141)$$

Dependiendo de los valores de slice\_type, collocated\_from\_10\_flag y collocated\_ref\_idx, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden obtener la variable colPic, que especifica la imagen que contiene la partición coubicada, como sigue:

- Si slice\_type es igual a B y collocated\_from\_10\_flag es igual a 0, la variable colPic especifica la imagen que contiene la partición coubicada según lo especificado por RefPicList1[collocated\_ref\_idx].

- De lo contrario (slice\_type es igual a B y collocated\_from\_10\_flag es igual a 1 o slice\_type es igual a P), la variable colPic especifica la imagen que contiene la partición coubicada según lo especificado por RefPicList0[collocated\_ref\_idx].

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden obtener la variable colPu y su posición (xPCol, yPCol) usando las siguientes etapas ordenadas, donde el texto subrayado representa cambios con respecto a HEVC WD7:

1. La variable colPu se puede obtener como sigue

$$yPRb = yP + nPSH \quad (8-139)$$

- 5 - Si  $(yP \gg \text{Log2CtbSize})$  es igual a  $(yPRb \gg \text{Log2CtbSize})$ , la componente horizontal de la posición de luma inferior derecha de la unidad de predicción actual se define mediante

$$xPRb = xP + nPSW \quad (8-140)$$

- 10 y la variable colPu se establece como la unidad de predicción que cubre la posición modificada dada por  $((xPRb \gg 4) \ll 4, (yPRb \gg 4) \ll 4)$  dentro de colPic.

- De lo contrario  $((yP \gg \text{Log2CtbSize})$  no es igual a  $(yPRb \gg \text{Log2CtbSize}))$ , colPu se marca como "no disponible".

- 15 2. Cuando colPu se codifica en un modo de intrapredicción o colPu se marca como "no disponible", se aplica lo siguiente.

- La posición de luma central de la unidad de predicción actual se define mediante

$$xPCtr = (xP + nPSW \gg 1) \quad (8-141)$$

$$yPCtr = (yP + nPSH \gg 1) \quad (8-142)$$

20 - La variable colPu se establece como la unidad de predicción que cubre la posición modificada dada por  $((xPCtr \gg 4) \ll 4, (yPCtr \gg 4) \ll 4)$  dentro de colPic.

- 25 3.  $(xPCol, yPCol)$  se establece igual a la muestra de luma superior izquierda de colPu en relación con la muestra de luma superior izquierda de colPic.

La función LongTermRefPic(picX, reldx, LX) se define como sigue. Si la imagen de referencia con índice reldx de la lista de imágenes de referencia LX de la imagen picX se marcó como "usada como referencia a largo plazo" en el momento en que picX era la imagen actual, LongTermRefPic(picX, reldx, LX) devuelve 1; de lo contrario, LongTermRefPic(picX, reldx, LX) devuelve 0.

La función AddPicId(picX, reldx, LX) devuelve AddPicId(pic), en la que pic es la imagen de referencia con índice reldx de la lista de imágenes de referencia LX de la imagen picX.

35 Las variables mvLXCol y availableFlagLXCol se obtienen como sigue.

- Si se cumple una o más de las siguientes condiciones, ambas componentes de mvLXCol se establecen igual a 0 y availableFlagLXCol se establece igual a 0.

40 - colPu se codifica en un modo de intrapredicción.

- colPu se marca como "no disponible".

45 - pic\_temporal\_mvp\_enable\_flag es igual a 0.

- De lo contrario, el vector de movimiento mvCol, el índice de referencia reldxCol y el identificador de lista de referencia listCol se obtienen como sigue.

50 - Si PredFlagL0[xPCol][yPCol] es igual a 0, mvCol, reldxCol y listCol se establecen igual a MvL1[xPCol][yPCol], RefldxL1[xPCol][yPCol] y L1, respectivamente.

- De lo contrario (PredFlagL0[xPCol][yPCol] es igual a 1), se aplica lo siguiente.

55 - Si PredFlagL1[xPCol][yPCol] es igual a 0, mvCol, reldxCol y listCol se establecen igual a MvL0[xPCol][yPCol], RefldxL0[xPCol][yPCol] y L0, respectivamente.

- De lo contrario (PredFlagL1[xPCol][yPCol] es igual a 1), se realizan las siguientes asignaciones.

- Si  $\text{PicOrderCnt}(\text{pic})$  de cada imagen pic en cada lista de imágenes de referencia es menor que o igual a  $\text{PicOrderCntVal}$ ,  $\text{mvCol}$ ,  $\text{refldxCOL}$  y  $\text{listCol}$  se establecen igual a  $\text{MvLX}[\text{xPCOL}][\text{yPCOL}]$ ,  $\text{RefldXLX}[\text{xPCOL}][\text{yPCOL}]$  y  $\text{LX}$ , respectivamente, siendo X el valor de X para el que se invoca este proceso.

5 - De lo contrario ( $\text{PicOrderCnt}(\text{pic})$  de al menos una imagen pic en al menos una lista de imágenes de referencia es mayor que  $\text{PicOrderCntVal}$ ),  $\text{mvCol}$ ,  $\text{refldxCOL}$  y  $\text{listCol}$  se establecen igual a  $\text{MvLN}[\text{xPCOL}][\text{yPCOL}]$ ,  $\text{RefldxLN}[\text{xPCOL}][\text{yPCOL}]$  y  $\text{LN}$ , respectivamente, siendo N el valor de  $\text{collocated\_from\_10\_flag}$ .

- Si se cumple una de las siguientes condiciones, la variable  $\text{availableFlagLXCol}$  se establece igual a 0.

10 -  $\text{AddPicId}(\text{RefPicListX}[\text{refldXLX}])$  no es igual a  $\text{AddPicId}(\text{colPic}, \text{refldxCOL}, \text{listCol})$ ;  
-  $\text{RefPicListX}[\text{refldXLX}]$  es una imagen de referencia a corto plazo y  $\text{LongTermRefPic}(\text{colPic}, \text{refldxCOL}, \text{listCol})$  es igual a 1;

15 -  $\text{RefPicListX}[\text{refldXLX}]$  es una imagen de referencia a largo plazo y  $\text{LongTermRefPic}(\text{colPic}, \text{refldxCOL}, \text{listCol})$  es igual a 0;

20 -  $\text{RefPicListX}[\text{refldXLX}]$  es una imagen de referencia a largo plazo y  $\text{LongTermRefPic}(\text{colPic}, \text{refldxCOL}, \text{listCol})$  es igual a 1 y  $\text{RefPicOrderCnt}(\text{colPic}, \text{refldxCOL}, \text{listCol})$  no es igual a  $\text{PicOrderCnt}(\text{RefPicListX}[\text{refldXLX}])$ .

- De lo contrario, la variable  $\text{availableFlagLXCol}$  se establece igual a 1 y se aplica lo siguiente.

25 - Si  $\text{RefPicListX}[\text{refldXLX}]$  es una imagen de referencia a largo plazo, o  $\text{LongTermRefPic}(\text{colPic}, \text{refldxCOL}, \text{listCol})$  es igual a 1, o  $\text{PicOrderCnt}(\text{colPic}) - \text{RefPicOrderCnt}(\text{colPic}, \text{refldxCOL}, \text{listCol})$  es igual a  $\text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListX}[\text{refldXLX}])$ ,

$$\text{mvLXCol} = \text{mvCol} \quad (8-143)$$

30 - De lo contrario,  $\text{mvLXCol}$  se obtiene como una versión escalada del vector de movimiento  $\text{mvCol}$  como se especifica a continuación.

$$\text{tx} = (16384 + (\text{Abs}(\text{td}) \gg 1)) / \text{td} \quad (8-144)$$

$$\text{DistScaleFactor} = \text{Clip3}(-4096, 4095, (\text{tb} * \text{tx} + 32) \gg 6) \quad (8-145)$$

$$\text{mvLXCol} =$$

$$\text{Clip3}(-8192, 8191.75, \text{Sign}(\text{DistScaleFactor} * \text{mvCol}) * ((\text{Abs}(\text{DistScaleFactor} * \text{mvCol}) + 127) \gg 8)) \quad (8-146)$$

35 - donde  $\text{td}$  y  $\text{tb}$  pueden obtenerse como

$$\text{td} = \text{Clip3}(-128, 127, \text{PicOrderCntVal}(\text{colPic}) - \text{RefPicOrderCnt}(\text{colPic}, \text{refldxCOL}, \text{listCol})) \quad (8-147)$$

$$\text{tb} = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListX}[\text{refldXLX}])) \quad (8-148)$$

40 El codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para realizar el siguiente proceso de obtención para candidatos combinados de fusión bipredictiva. Los datos de entrada del proceso pueden incluir una lista de candidatos de fusión  $\text{mergeCandList}$ , índices de referencia  $\text{refldxL0N}$  y  $\text{refldxL1N}$  de cada candidato N que está en  $\text{mergeCandList}$ , banderas de utilización de lista de predicción  $\text{predFlagL0N}$  y  $\text{predFlagL1N}$  de cada candidato N que está en  $\text{mergeCandList}$ , vectores de movimiento  $\text{mvL0N}$  y  $\text{mvL1N}$  de cada candidato N que está en  $\text{mergeCandList}$ , el número de elementos  $\text{numMergeCand}$  dentro de  $\text{mergeCandList}$  y el número de elementos  $\text{numOrigMergeCand}$  dentro de  $\text{mergeCandList}$  después del proceso de obtención de candidato de fusión espacial y temporal. Los datos de salida de este proceso pueden incluir la lista de candidatos de fusión  $\text{mergeCandList}$ , el número de elementos  $\text{numMergeCand}$  dentro de  $\text{mergeCandList}$ , índices de referencia  $\text{refldxL0combCandk}$  y  $\text{refldxL1combCandk}$  de cada nuevo candidato  $\text{combCandk}$  que se añade en  $\text{mergeCandList}$  durante la invocación de este proceso, banderas de utilización de lista de predicción  $\text{predFlagL0combCandk}$  y  $\text{predFlagL1combCandk}$  de cada nuevo candidato  $\text{combCandk}$  que se añade en  $\text{mergeCandList}$  durante la invocación de este proceso, y vectores de movimiento  $\text{mvL0combCandk}$  y  $\text{mvL1combCandk}$  de cada nuevo candidato  $\text{combCandk}$  que se añaden en  $\text{mergeCandList}$  durante la invocación de este proceso.

Cuando numOrigMergeCand es mayor que 1 y menor que MaxNumMergeCand, la variable numInputMergeCand puede establecerse en numMergeCand, las variables combIdx y combCnt pueden establecerse en 0, la variable combStop puede establecerse en FALSE y las siguientes etapas pueden repetirse hasta que combStop sea igual a TRUE (donde las elipsis representan las mismas etapas que se proporcionan en HEVC WD7, y el texto subrayado representa cambios con respecto a HEVC WD7):

1. Las variables l0CandIdx e l1CandIdx se obtienen usando combIdx como se especifica en la Tabla 8-8.

2. Las siguientes asignaciones se realizan con l0Cand como candidato en la posición l0CandIdx e l1Cand como el candidato en la posición l1CandIdx en la lista de candidatos de fusión mergeCandList (l0Cand = mergeCandList[l0CandIdx], l1Cand = mergeCandList[l1CandIdx]).

3. Cuando se cumplen todas las condiciones siguientes,

- predFlagL0l0Cand == 1

- predFlagL1l1Cand == 1

-  $\frac{\text{AddPicId}(\text{RefPicListL0}[\text{refIdxL0l0Cand}])}{\text{PicOrderCnt}(\text{RefPicList0}[\text{refIdxL0l0Cand}])} \neq \frac{\text{AddPicId}(\text{RefPicListL1}[\text{refIdxL1l1Cand}])}{\text{PicOrderCnt}(\text{RefPicList1}[\text{refIdxL1l1Cand}])} \mid \text{mvL0l0Cand} \neq \text{mvL1l1C}$  y se aplica lo siguiente.

4. ...

5. ...

Como una alternativa que no pertenece a la invención, la predicción entre dos imágenes de referencia a largo plazo puede habilitarse sin escalado, y la predicción entre dos imágenes de referencia entre vistas puede habilitarse sin escalado. El codificador de vídeo 20 y el descodificador de vídeo 30 pueden configurarse para realizar un proceso de obtención para candidatos a predictor de vector de movimiento como sigue, donde el texto subrayado representa cambios con respecto a HEVC WD7 y las elipsis representan texto que es idéntico al de HEVC WD7:

La variable isScaledFlagLX, donde X es 0 o 1, puede establecerse igual a 0.

El vector de movimiento mvLXA y la bandera de disponibilidad availableFlagLXA pueden obtenerse en las siguientes etapas ordenadas:

1. ...

2. ...

3. ...

4. ...

5. Cuando availableFlagLXA es igual a 0, para (xA<sub>k</sub>, yA<sub>k</sub>) desde (xA<sub>0</sub>, yA<sub>0</sub>) hasta (xA<sub>1</sub>, yA<sub>1</sub>) donde yA<sub>1</sub> = yA<sub>0</sub> - MinPSize, lo siguiente se aplica repetidamente hasta que availableFlagLXA sea igual a 1:

- Si la unidad de predicción que cubre la ubicación de luma (xA<sub>k</sub>, yA<sub>k</sub>) está disponible [Ed. (BB): se reescribe usando MinCbAddrZS[ ] y el proceso de disponibilidad para bloques de codificación mínimos], PredMode no es MODE\_INTRA, predFlagLY[xA<sub>k</sub>][yA<sub>k</sub>] es igual a 1 y RefPicType(RefPicListX[refIdxLX]) es igual a RefPicType(RefPicListX[refIdxLX[xA<sub>k</sub>][yA<sub>k</sub>]]), availableFlagLXA se establece igual a 1, el vector de movimiento mvLXA se establece igual al vector de movimiento mvLX[xA<sub>k</sub>][yA<sub>k</sub>], refIdxA se establece igual a refIdxLX[xA<sub>k</sub>][yA<sub>k</sub>], ListA se establece igual a ListX.

- De lo contrario, si la unidad de predicción que cubre la ubicación de luma (xA<sub>k</sub>, yA<sub>k</sub>) está disponible [Ed. (BB): se reescribe usando MinCbAddrZS[ ] y el proceso de disponibilidad para bloques de codificación mínimos], PredMode no es MODE\_INTRA, predFlagLY[xA<sub>k</sub>][yA<sub>k</sub>] (con Y ≠ X) es igual a 1, y RefPicType(RefPicListX[refIdxLX]) es igual a RefPicType(RefPicListY[refIdxLY[xA<sub>k</sub>][yA<sub>k</sub>]]), availableFlagLXA se establece igual a 1, el vector de movimiento mvLXA se establece igual al vector de movimiento mvLY[xA<sub>k</sub>][yA<sub>k</sub>], refIdxA se establece igual a refIdxLY[xA<sub>k</sub>][yA<sub>k</sub>], ListA se establece igual a ListY.

- Cuando availableFlagLXA es igual a 1, y tanto RefPicListA[refIdxA] como RefPicListX[refIdxLX] son imágenes de referencia a corto plazo, mvLXA se obtiene como se especifica a continuación.

$$tx = (16384 + (Abs(td) \gg 1))/td \quad (8-126)$$

$$DistScaleFactor = Clip3(-4096, 4095, (tb * tx + 32) \gg 6) \quad (8-127)$$

$$mvLXA = Clip3(-8192, 8191.75, Sign(DistScaleFactor * mvLXA) * ((Abs(DistScaleFactor * mvLXA) + 127) \gg 8)) \quad (8-128)$$

donde td y tb pueden obtenerse como

$$td = Clip3(-128, 127, PicOrderCntVal - PicOrderCnt(RefPicListA[refIdxA])) \quad (8-129)$$

$$td = Clip3(-128, 127, PicOrderCntVal - PicOrderCnt(RefPicListX[refIdxLX])) \quad (8-130)$$

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden obtener el vector de movimiento mvLXB y la bandera de disponibilidad availableFlagLXB usando las siguientes etapas ordenadas, donde el texto subrayado representa cambios con respecto a HEVC WD7 y las elipsis representan texto que es idéntico al de HEVC WD7:

1. ...

2. ...

3. ...

4. ...

5. ...

6. Cuando isScaledFlagLX es igual a 0, availableFlagLXB se establece igual a 0 y para (xB<sub>k</sub>, yB<sub>k</sub>) desde (xB<sub>0</sub>, yB<sub>0</sub>) hasta (xB<sub>2</sub>, yB<sub>2</sub>) donde xB<sub>0</sub> = xP + nPSW, xB<sub>1</sub> = xB<sub>0</sub> - MinPuSize, y xB<sub>2</sub> = xP - MinPuSize, lo siguiente se aplica repetidamente hasta que availableFlagLXB sea igual a 1:

- Si la unidad de predicción que cubre la ubicación de luma (xB<sub>k</sub>, yB<sub>k</sub>) está disponible [Ed. (BB): se reescribe usando MinCbAddrZS[ ][ ] y el proceso de disponibilidad para bloques de codificación mínimos], PredMode no es MODE\_INTRA, predFlagLX[xB<sub>k</sub>][yB<sub>k</sub>] es igual a 1 y RefPicType(RefPicListX[refIdxLX]) es igual a RefPicType(RefPicListX[refIdxLX][xB<sub>k</sub>][yB<sub>k</sub>]), availableFlagLXB se establece igual a 1, el vector de movimiento mvLXB se establece igual al vector de movimiento mvLX[xB<sub>k</sub>][yB<sub>k</sub>], refIdxB se establece igual a refIdxLX[xB<sub>k</sub>][yB<sub>k</sub>], ListB se establece igual a ListX.

- De lo contrario, si la unidad de predicción que cubre la ubicación de luma (xB<sub>k</sub>, yB<sub>k</sub>) está disponible [Ed. (BB): se reescribe usando MinCbAddrZS[ ][ ] y el proceso de disponibilidad para bloques de codificación mínimos], PredMode no es MODE\_INTRA, predFlagLY[xB<sub>k</sub>][yB<sub>k</sub>] (con Y ≠ X) es igual a 1, y RefPicType(RefPicListX[refIdxLX]) es igual a RefPicType(RefPicListY[refIdxLY][xB<sub>k</sub>][yB<sub>k</sub>]), availableFlagLXB se establece igual a 1, el vector de movimiento mvLXB se establece igual al vector de movimiento mvLY[xB<sub>k</sub>][yB<sub>k</sub>], refIdxB se establece igual a refIdxLY[xB<sub>k</sub>][yB<sub>k</sub>], ListB se establece igual a ListY.

- Cuando availableFlagLXB es igual a 1 y PicOrderCnt(RefPicListB[refIdxB]) no es igual a PicOrderCnt(RefPicListX[refIdxLX]) y tanto RefPicListB[refIdxB] como RefPicListX[refIdxLX] son imágenes de referencia de corto plazo, mvLXB se obtiene como se especifica a continuación.

$$tx = (16384 + (Abs(td) \gg 1))/td \quad (8-134)$$

$$DistScaleFactor = Clip3(-4096, 4095, (tb * tx + 32) \gg 6) \quad (8-135)$$

$$mvLXB = Clip3(-8192, 8191.75, Sign(DistScaleFactor * mvLXA) * ((Abs(DistScaleFactor * mvLXA) + 127) \gg 8)) \quad (8-136)$$

donde td y tb pueden obtenerse como

$$td = Clip3(-128, 127, PicOrderCntVal - PicOrderCnt(RefPicListB[refIdxB])) \quad (8-137)$$

$$td = Clip3(-128, 127, PicOrderCntVal - PicOrderCnt(RefPicListX[refIdxLX])) \quad (8-138)$$

El codificador de vídeo 20 y el descodificador de vídeo 30 pueden configurarse para realizar un proceso de obtención para la predicción temporal de vectores de movimiento de luma, como sigue.

Las variables mvLXCol y availableFlagLXCol pueden obtenerse como sigue.

- Si se cumple una o más de las siguientes condiciones, ambas componentes de mvLXCol se establecen igual a 0 y availableFlagLXCol se establece igual a 0.

- colPu se codifica en un modo de intrapredicción.

- colPu se marca como "no disponible".

- pic\_temporal\_mvp\_enable\_flag es igual a 0.

- De lo contrario, el vector de movimiento mvCol, el índice de referencia refIdxCol y el identificador de lista de referencia listCol se obtienen como sigue.

- Si PredFlagL0[xPCol][yPCol] es igual a 0, mvCol, refIdxCol y listCol se establecen igual a MvL1[xPCol][yPCol], RefIdxL1[xPCol][yPCol] y L1, respectivamente.

- De lo contrario (PredFlagL0[xPCol][yPCol] es igual a 1), se aplica lo siguiente.

- Si PredFlagL1[xPCol][yPCol] es igual a 0, mvCol, refIdxCol y listCol se establecen igual a MvL0[xPCol][yPCol], RefIdxL0[xPCol][yPCol] y L0, respectivamente.

- De lo contrario (PredFlagL1[xPCol][yPCol] es igual a 1), se realizan las siguientes asignaciones.

- Si PicOrderCnt(pic) de cada imagen pic en cada lista de imágenes de referencia es menor que o igual a PicOrderCntVal, mvCol, refIdxCol y listCol se establecen igual a MvLX[xPCol][yPCol], RefIdxLX[xPCol][yPCol] y LX, respectivamente, siendo X el valor de X para el que se invoca este proceso.

- De lo contrario (PicOrderCnt(pic) de al menos una imagen pic en al menos una lista de imágenes de referencia es mayor que PicOrderCntVal), mvCol, refIdxCol y listCol se establecen igual a MvLN[xPCol][yPCol], RefIdxLN[xPCol][yPCol] y LN, respectivamente, siendo N el valor de collocated\_from\_l0\_flag.

- Si RefPicType(RefPicListX[refIdxLX]) no es igual a RefPicType(colPic, refIdxCol, listCol), la variable availableFlagLXCol se establece igual a 0.

- De lo contrario, la variable availableFlagLXCol se establece igual a 1 y se aplica lo siguiente.

- Si RefPicListX[refIdxLX] es una imagen de referencia a largo plazo, o LongTermRefPic(colPic, refIdxCol, listCol) es igual a 1, o PicOrderCnt(colPic) - RefPicOrderCnt(colPic, refIdxCol, listCol) es igual a PicOrderCntVal - PicOrderCnt(RefPicListX[refIdxLX]),

$$mvLXCol = mvCol \text{ (8-143)}$$

- De lo contrario, mvLXCol se obtiene como una versión escalada del vector de movimiento mvCol como se especifica a continuación.

$$tx = (16384 + (Abs(td) \gg 1)) / td \text{ (8-144)}$$

$$DistScaleFactor = Clip3(-4096, 4095, (tb * tx + 32) \gg 6) \text{ (8-145)}$$

$$mvLXCol =$$

$$Clip3(-8192, 8191.75, Sign(DistScaleFactor * mvCol) * ((Abs(DistScaleFactor * mvCol) + 127) \gg 8)) \text{ (8-146)}$$

donde td y tb pueden obtenerse como

$$td = Clip3(-128, 127, PicOrderCntVal(colPic) - RefPicOrderCnt(colPic, refIdxCol, listCol)) \text{ (8-147)}$$

$$tb = Clip3(-128, 127, PicOrderCntVal - PicOrderCnt(RefPicListX[refIdxLX])) \text{ (8-148)}$$



La sección 8.5.2.1.3 de HEVC WD7 puede permanecer invariable para los propósitos de este ejemplo.

En un ejemplo alternativo que no pertenece a la invención, la predicción entre vectores de movimiento que hacen referencia a diferentes imágenes de referencia a largo plazo puede estar inhabilitada, la predicción entre vectores de movimiento que hacen referencia a una imagen de referencia entre vistas y a una imagen de referencia a largo plazo puede estar inhabilitada, y la predicción entre vectores de movimiento que hacen referencia a diferentes imágenes de referencia entre vistas puede estar habilitada. En este ejemplo, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden configurarse para realizar un proceso de obtención para candidatos a predictor de vector de movimiento como se describe a continuación. Los datos de entrada para este proceso pueden incluir una ubicación de luma ( $x_P$ ,  $y_P$ ) que especifica la muestra de luma superior izquierda de la unidad de predicción actual en relación con la muestra superior izquierda de la imagen actual, variables que especifican la anchura y la altura de la unidad de predicción para luma,  $n_{PSW}$  y  $n_{PSH}$ , y el índice de referencia de la partición de unidad de predicción actual  $refIdxLX$  (siendo  $X$  0 o 1). Los datos de salida de este proceso pueden incluir (donde  $N$  se reemplaza por  $A$  o  $B$ ) los vectores de movimiento  $mvLXN$  de las unidades de predicción vecinas y las banderas de disponibilidad  $availableFlagLXN$  de las unidades de predicción vecinas.

La variable  $isScaledFlagLX$ , donde  $X$  es 0 o 1, puede establecerse igual a 0. El codificador de vídeo 20 y el descodificador de vídeo 30 pueden obtener el vector de movimiento  $mvLXA$  y la bandera de disponibilidad  $availableFlagLXA$  usando las siguientes etapas ordenadas, donde el texto subrayado representa cambios con respecto a HEVC WD7:

1. Sea un conjunto de dos ubicaciones de muestra  $(x_{A_k}, y_{A_k})$ , con  $k = 0, 1$ , que especifica ubicaciones de muestra con  $x_{A_k} = x_P - 1$ ,  $y_{A_0} = y_P + n_{PSH}$  e  $y_{A_1} = y_{A_0} - \text{MinPuSize}$ . El conjunto de ubicaciones de muestra  $(x_{A_k}, y_{A_k})$  representa las ubicaciones de muestra justo en el lado izquierdo del límite de partición izquierdo y su línea extendida.

2. Establézcase la bandera de disponibilidad  $availableFlagLXA$  inicialmente igual 0 y ambas componentes de  $mvLXA$  igual a 0.

3. Cuando una o más de las siguientes condiciones son verdaderas, la variable  $isScaledFlagLX$  se establece igual a 1.

- La unidad de predicción que cubre la ubicación de luma  $(x_{A_0}, y_{A_0})$  está disponible [Ed. (BB): se reescribe usando  $\text{MinCbAddrZS}[ ][ ]$  y el proceso de disponibilidad para bloques de codificación mínimos] y  $\text{PredMode}$  no es  $\text{MODE\_INTRA}$ .

- La unidad de predicción que cubre la ubicación de luma  $(x_{A_1}, y_{A_1})$  está disponible [Ed. (BB): se reescribe usando  $\text{MinCbAddrZS}[ ][ ]$  y el proceso de disponibilidad para bloques de codificación mínimos] y  $\text{PredMode}$  no es  $\text{MODE\_INTRA}$ .

4. Para  $(x_{A_k}, y_{A_k})$  de  $(x_{A_0}, y_{A_0})$  a  $(x_{A_1}, y_{A_1})$  donde  $y_{A_1} = y_{A_0} - \text{MinPuSize}$ , si  $availableFlagLXA$  es igual a 0, se aplica lo siguiente:

- Si la unidad de predicción que cubre la ubicación de luma  $(x_{A_k}, y_{A_k})$  está disponible [Ed. (BB): se reescribe usando  $\text{MinCbAddrZS}[ ][ ]$  y el proceso de disponibilidad para bloques de codificación mínimos],  $\text{PredMode}$  no es  $\text{MODE\_INTRA}$ ,  $\text{predFlagLX}[x_{A_k}][y_{A_k}]$  es igual a 1 y el índice de referencia  $\text{refIdxLX}[x_{A_k}][y_{A_k}]$  es igual al índice de referencia de la unidad de predicción actual  $\text{refIdxLX}$ ,  $availableFlagLXA$  se establece igual a 1 y el vector de movimiento  $mvLXA$  se establece igual al vector de movimiento  $mvLX[x_{A_k}][y_{A_k}]$ ,  $\text{refIdxA}$  se establece igual a  $\text{refIdxLX}[x_{A_k}][y_{A_k}]$  y  $\text{ListA}$  se establece igual a  $\text{ListX}$ .

- De lo contrario, si la unidad de predicción que cubre la ubicación de luma  $(x_{A_k}, y_{A_k})$  está disponible [Ed. (BB): se reescribe usando  $\text{MinCbAddrZS}[ ][ ]$  y el proceso de disponibilidad para bloques de codificación mínimos],  $\text{PredMode}$  no es  $\text{MODE\_INTRA}$ ,  $\text{predFlagLY}[x_{A_k}][y_{A_k}]$  (con  $Y \neq X$ ) es igual a 1,  $\text{AddPicId}(\text{RefPicListX}[\text{refIdxLX}])$  es igual a  $\text{AddPicId}(\text{RefPicListY}[\text{refIdxLY}[x_{A_k}][y_{A_k}]])$ , y  $\text{PicOrderCnt}(\text{RefPicListY}[\text{refIdxLY}[x_{A_k}][y_{A_k}]])$  es igual a  $\text{PicOrderCnt}(\text{RefPicListX}[\text{refIdxLX}])$ ,  $availableFlagLXA$  se establece igual a 1, el vector de movimiento  $mvLXA$  se establece igual al vector de movimiento  $mvLY[x_{A_k}][y_{A_k}]$ ,  $\text{refIdxA}$  se establece igual a  $\text{refIdxLY}[x_{A_k}][y_{A_k}]$ ,  $\text{ListA}$  se establece igual a  $\text{ListY}$  y  $mvLXA$  se establece igual a  $mvLXA$ .

- Cuando  $availableFlagLXA$  es igual a 1,  $availableFlagLXA$  se establece en 0 si se cumple lo siguiente

- Uno y solo uno de  $\text{RefPicListX}[\text{refIdxLX}]$  y  $\text{ListA}[\text{refIdxA}]$  es una imagen de referencia a largo plazo.

5. Cuando  $availableFlagLXA$  es igual a 0, para  $(x_{A_k}, y_{A_k})$  de  $(x_{A_0}, y_{A_0})$  a  $(x_{A_1}, y_{A_1})$  donde  $y_{A_1} = y_{A_0} - \text{MinPuSize}$ , si  $availableFlagLXA$  es igual a 0, se aplica lo siguiente:

- Si la unidad de predicción que cubre la ubicación de luma ( $x_{A_k}$ ,  $y_{A_k}$ ) está disponible [Ed. (BB): se reescribe usando MinCbAddrZS[ ][ ] y el proceso de disponibilidad para bloques de codificación mínimos], PredMode no es MODE\_INTRA, predFlagLX[ $x_{A_k}$ ][ $y_{A_k}$ ] es igual a 1, y AddPicId(RefPicListLX[refIdxLX]) es igual a AddPicId(RefPicListLX[refIdxLX][ $x_{A_k}$ ][ $y_{A_k}$ ]), availableFlagLXA se establece igual a 1, el vector de movimiento mvLXA se establece igual al vector de movimiento mvLX[ $x_{A_k}$ ][ $y_{A_k}$ ], refIdxA se establece igual a refIdxLX[ $x_{A_k}$ ][ $y_{A_k}$ ], ListA se establece igual a ListX.

- De lo contrario, si la unidad de predicción que cubre la ubicación de luma ( $x_{A_k}$ ,  $y_{A_k}$ ) está disponible [Ed. (BB): se reescribe usando MinCbAddrZS[ ][ ] y el proceso de disponibilidad para bloques de codificación mínimos], PredMode no es MODE\_INTRA, predFlagLY[ $x_{A_k}$ ][ $y_{A_k}$ ] (con  $Y \neq X$ ) es igual a 1, y AddPicId(RefPicListLX[refIdxLX]) es igual a AddPicId(RefPicListLY[refIdxLY[ $x_{A_k}$ ][ $y_{A_k}$ ]]), availableFlagLXA se establece igual a 1, el vector de movimiento mvLXA se establece igual al vector de movimiento mvLY[ $x_{A_k}$ ][ $y_{A_k}$ ], refIdxA se establece igual a refIdxLY[ $x_{A_k}$ ][ $y_{A_k}$ ], ListA se establece igual a ListY.

- Cuando availableFlagLXA es igual a 1, availableFlagLXA se establece en 0 si se cumple lo siguiente:

- Uno y solo uno de RefPicListX[refIdxLX] y ListA[refIdxA] es una imagen de referencia a largo plazo.

- Cuando availableFlagLXA es igual a 1, y tanto RefPicListA[refIdxA] como RefPicListX[refIdxLX] son imágenes de referencia a corto plazo, mvLXA se obtiene como se especifica a continuación.

$$tx = (16384 + (\text{Abs}(td) \gg 1)) / td \quad (8-126)$$

$$\text{DistScaleFactor} = \text{Clip3}(-4096, 4095, (tb * tx + 32) \gg 6) \quad (8-127)$$

$$mvLXA = \text{Clip3}(-8192, 8191.75, \text{Sign}(\text{DistScaleFactor} * mvLXA) * ((\text{Abs}(\text{DistScaleFactor} * mvLXA) + 127) \gg 8)) \quad (8-128)$$

donde td y tb pueden obtenerse como

$$td = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListA}[\text{refIdxA}])) \quad (8-129)$$

$$tb = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListA}[\text{refIdxA}])) \quad (8-130)$$

El codificador de vídeo 20 y el descodificador de vídeo 30 pueden configurarse para obtener el vector de movimiento mvLXB y la bandera de disponibilidad availableFlagLXB usando las siguientes etapas ordenadas, donde el texto subrayado representa cambios con respecto a HEVC WD7:

1. Sea un conjunto de tres ubicaciones de muestra ( $x_{B_k}$ ,  $y_{B_k}$ ), con  $k = 0, 1, 2$ , que especifica ubicaciones de muestra con  $x_{B_0} = x_P + nPSW$ ,  $x_{B_1} = x_{B_0} - \text{MinPuSize}$ ,  $x_{B_2} = x_P - \text{MinPuSize}$  e  $y_{B_k} = y_P - 1$ . El conjunto de ubicaciones de muestra ( $x_{B_k}$ ,  $y_{B_k}$ ) representa las ubicaciones de muestra justo en el lado superior del límite de partición superior y su línea extendida. [Ed. (BB): se define MinPuSize en el SPS, pero la obtención debe depender del uso de una bandera AMP]

2. Cuando  $y_P - 1$  es menor que  $((y_C \gg \text{Log2CtbSize}) \ll \text{Log2CtbSize})$ , se aplica lo siguiente.

$$x_{B_0} = (x_{B_0} \gg 3) \ll 3 + ((x_{B_0} \gg 3) \& 1) * 7 \quad (8-131)$$

$$x_{B_1} = (x_{B_1} \gg 3) \ll 3 + ((x_{B_1} \gg 3) \& 1) * 7 \quad (8-132)$$

$$x_{B_2} = (x_{B_2} \gg 3) \ll 3 + ((x_{B_2} \gg 3) \& 1) * 7 \quad (8-133)$$

3. Establézcase la bandera de disponibilidad availableFlagLXB inicialmente igual 0 y ambas componentes de mvLXB igual a 0.

4. Para ( $x_{B_k}$ ,  $y_{B_k}$ ) desde ( $x_{B_0}$ ,  $y_{B_0}$ ) hasta ( $x_{B_2}$ ,  $y_{B_2}$ ), donde  $x_{B_0} = x_P + nPSW$ ,  $x_{B_1} = x_{B_0} - \text{MinPuSize}$  y  $x_{B_2} = x_P - \text{MinPuSize}$ , si availableFlagLXB es igual a 0, se aplica lo siguiente:

- Si la unidad de predicción que cubre la ubicación de luma ( $x_{B_k}$ ,  $y_{B_k}$ ) está disponible [Ed. (BB): se reescribe usando MinCbAddrZS[ ][ ] y el proceso de disponibilidad para bloques de codificación mínimos], PredMode no es MODE\_INTRA, predFlagLX[ $x_{B_k}$ ][ $y_{B_k}$ ] es igual a 1 y el índice de referencia refIdxLX[ $x_{B_k}$ ][ $y_{B_k}$ ] es igual al índice de referencia de la unidad de predicción actual refIdxLX, availableFlagLXB se establece igual a 1 y el vector de movimiento mvLXB se establece igual al vector de movimiento mvLX[ $x_{B_k}$ ][ $y_{B_k}$ ], refIdxB se establece igual a refIdxLX[ $x_{B_k}$ ][ $y_{B_k}$ ] y ListB se establece igual a ListX.

- De lo contrario, si la unidad de predicción que cubre la ubicación de luma ( $x_{B_k}$ ,  $y_{B_k}$ ) está disponible [Ed. (BB): se reescribe usando  $\text{MinCbAddrZS}[\ ]$  y el proceso de disponibilidad para bloques de codificación mínimos],  $\text{PredMode}$  no es  $\text{MODE\_INTRA}$ ,  $\text{predFlagLY}[x_{B_k}][y_{B_k}]$  (con  $Y \neq X$ ) es igual a 1,  $\text{AddPicId}(\text{RefPicListX}[\text{refldxLX}])$  es igual a  $\text{AddPicId}(\text{RefPicListLY}[\text{refldxLY}[x_{B_k}][y_{B_k}]])$ , y  $\text{PicOrderCnt}(\text{RefPicListY}[\text{refldxLY}[x_{B_k}][y_{B_k}]])$  es igual a  $\text{PicOrderCnt}(\text{RefPicListX}[\text{refldxLX}])$ ,  $\text{availableFlagLXB}$  se establece igual a 1, el vector de movimiento  $\text{mvLXB}$  se establece igual al vector de movimiento  $\text{mvLY}[x_{B_k}][y_{B_k}]$ ,  $\text{refldxB}$  se establece igual a  $\text{refldxLY}[x_{B_k}][y_{B_k}]$ , y  $\text{ListB}$  se establece igual a  $\text{ListY}$ .

- Cuando  $\text{availableFlagLXA}$  es igual a 1,  $\text{availableFlagLXA}$  se establece en 0 si se cumple lo siguiente:

- Uno y solo uno de  $\text{RefPicListX}[\text{refldxLX}]$  y  $\text{ListB}[\text{refldxB}]$  es una imagen de referencia a largo plazo.

5. Cuando  $\text{isScaledFlagLX}$  es igual a 0 y  $\text{availableFlagLXB}$  es igual a 1,  $\text{mvLXA}$  se establece igual a  $\text{mvLXB}$  y  $\text{refldxA}$  se establece igual a  $\text{refldxB}$  y  $\text{availableFlagLXA}$  se establece igual a 1.

6. Cuando  $\text{isScaledFlagLX}$  es igual a 0,  $\text{availableFlagLXB}$  se establece igual a 0 y para ( $x_{B_k}$ ,  $y_{B_k}$ ) desde ( $x_{B_0}$ ,  $y_{B_0}$ ) hasta ( $x_{B_2}$ ,  $y_{B_2}$ ) donde  $x_{B_0} = x_P + n\text{PSW}$ ,  $x_{B_1} = x_{B_0} - \text{MinPuSize}$ , y  $x_{B_2} = x_P - \text{MinPuSize}$ , si  $\text{availableFlagLXB}$  es igual a 0, se aplica lo siguiente:

- Si la unidad de predicción que cubre la ubicación de luma ( $x_{B_k}$ ,  $y_{B_k}$ ) está disponible [Ed. (BB): se reescribe usando  $\text{MinCbAddrZS}[\ ]$  y el proceso de disponibilidad para bloques de codificación mínimos],  $\text{PredMode}$  no es  $\text{MODE\_INTRA}$ ,  $\text{predFlagLX}[x_{B_k}][y_{B_k}]$  es igual a 1 y  $\text{AddPicId}(\text{RefPicListX}[\text{refldxLX}])$  es igual a  $\text{AddPicId}(\text{RefPicListX}[\text{refldxLX}[x_{B_k}][y_{B_k}]])$ ,  $\text{availableFlagLXB}$  se establece igual a 1, el vector de movimiento  $\text{mvLXB}$  se establece igual al vector de movimiento  $\text{mvLX}[x_{B_k}][y_{B_k}]$ ,  $\text{refldxB}$  se establece igual a  $\text{refldxLX}[x_{B_k}][y_{B_k}]$ ,  $\text{ListB}$  se establece igual a  $\text{ListX}$ .

- De lo contrario, si la unidad de predicción que cubre la ubicación de luma ( $x_{B_k}$ ,  $y_{B_k}$ ) está disponible [Ed. (BB): se reescribe usando  $\text{MinCbAddrZS}[\ ]$  y el proceso de disponibilidad para bloques de codificación mínimos],  $\text{PredMode}$  no es  $\text{MODE\_INTRA}$ ,  $\text{predFlagLY}[x_{B_k}][y_{B_k}]$  (con  $Y \neq X$ ) es igual a 1, y  $\text{AddPicId}(\text{RefPicListX}[\text{refldxLX}])$  es igual a  $\text{AddPicId}(\text{RefPicListY}[\text{refldxLY}[x_{B_k}][y_{B_k}]])$ ,  $\text{availableFlagLXB}$  se establece igual a 1, el vector de movimiento  $\text{mvLXB}$  se establece igual al vector de movimiento  $\text{mvLY}[x_{B_k}][y_{B_k}]$ ,  $\text{refldxB}$  se establece igual a  $\text{refldxLY}[x_{B_k}][y_{B_k}]$ ,  $\text{ListB}$  se establece igual a  $\text{ListY}$ .

- Cuando  $\text{availableFlagLXA}$  es igual a 1,  $\text{availableFlagLXA}$  se establece en 0 si se cumple lo siguiente:

- Uno y solo uno de  $\text{RefPicListX}[\text{refldxLX}]$  y  $\text{ListB}[\text{refldxB}]$  es una imagen de referencia a largo plazo.

- Cuando  $\text{availableFlagLXB}$  es igual a 1 y  $\text{PicOrderCnt}(\text{RefPicListB}[\text{refldxB}])$  no es igual a  $\text{PicOrderCnt}(\text{RefPicListX}[\text{refldxLX}])$  y tanto  $\text{RefPicListB}[\text{refldxB}]$  como  $\text{RefPicListX}[\text{refldxLX}]$  son imágenes de referencia de corto plazo,  $\text{mvLXB}$  se obtiene como se especifica a continuación.

$$tx = (16384 + (\text{Abs}(td) \gg 1)) / td \quad (8-134)$$

$$\text{DistScaleFactor} = \text{Clip3}(-4096, 4095, (tb * tx + 32) \gg 6) \quad (8-135)$$

$$\text{mvLXB} = \text{Clip3}(-8192, 8191.75, \text{Sign}(\text{DistScaleFactor} * \text{mvLXA}) * ((\text{Abs}(\text{DistScaleFactor} * \text{mvLXA}) + 127) \gg 8)) \quad (8-136)$$

donde  $td$  y  $tb$  pueden obtenerse como

$$td = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListB}[\text{refldxB}])) \quad (8-137)$$

$$tb = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListX}[\text{refldxLX}])) \quad (8-138)$$

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para implementar un proceso de obtención para la predicción temporal de vectores de movimiento de luma, como se analiza a continuación. Los datos de entrada para este proceso pueden incluir una ubicación de luma ( $x_P$ ,  $y_P$ ) que especifica la muestra de luma superior izquierda de la unidad de predicción actual en relación con la muestra superior izquierda de la imagen actual, variables que especifican la anchura y la altura de la unidad de predicción para luma,  $n\text{PSW}$  y  $n\text{PSH}$ , y el índice de referencia de la partición de unidad de predicción actual  $\text{refldxLX}$  (siendo  $X$  0 o 1). Los datos de salida de este proceso pueden incluir la predicción de vectores de movimiento  $\text{mvLXCol}$  y la bandera de disponibilidad  $\text{availableFlagLXCol}$ .

La función  $\text{RefPicOrderCnt}(\text{picX}, \text{refIdx}, \text{Lx})$ , en un ejemplo, devuelve el recuento de orden de imagen  $\text{PicOrderCntVal}$  de la imagen de referencia con índice  $\text{refIdx}$  de la lista de imágenes de referencia LX de la imagen  $\text{picX}$ , y se puede especificar como sigue:

$$\text{RefPicOrderCnt}(\text{picX}, \text{refIdx}, \text{LX}) = \text{PicOrderCnt}(\text{RefPicListX}[\text{refIdx}] \text{ de la imagen picX}) \quad (8-141)$$

Dependiendo de los valores de  $\text{slice\_type}$ ,  $\text{collocated\_from\_l0\_flag}$  y  $\text{collocated\_ref\_idx}$ , el codificador de vídeo 20 y el descodificador de vídeo 30 pueden obtener la variable  $\text{colPic}$ , que especifica la imagen que contiene la partición coubicada, como sigue:

- Si el tipo de segmento es igual a B y  $\text{collocated\_from\_l0\_flag}$  es igual a 0, la variable  $\text{colPic}$  especifica la imagen que contiene la partición coubicada según lo especificado por  $\text{RefPicList1}[\text{collocated\_ref\_idx}]$ .

- De lo contrario ( $\text{slice\_type}$  es igual a B y  $\text{collocated\_from\_l0\_flag}$  es igual a 1 o  $\text{slice\_type}$  es igual a P), la variable  $\text{colPic}$  especifica la imagen que contiene la partición coubicada según lo especificado por  $\text{RefPicList0}[\text{collocated\_ref\_idx}]$ .

El codificador de vídeo 20 y el descodificador de vídeo 30 pueden obtener la variable  $\text{colPu}$  y su posición ( $\text{xPCol}$ ,  $\text{yPCol}$ ) usando las siguientes etapas ordenadas:

1. La variable  $\text{colPu}$  se obtiene como sigue

$$\text{yPRb} = \text{yP} + \text{nPSH} \quad (8-139)$$

- Si ( $\text{yP} \gg \text{Log2CtbSize}$ ) es igual a ( $\text{yPRb} \gg \text{Log2CtbSize}$ ), la componente horizontal de la posición de luma inferior derecha de la unidad de predicción actual se define mediante

$$\text{xPRb} = \text{xP} + \text{nPSW} \quad (8-140)$$

y la variable  $\text{colPu}$  se establece como la unidad de predicción que cubre la posición modificada dada por  $((\text{xPRb} \gg 4) \ll 4, (\text{yPRb} \gg 4) \ll 4)$  dentro de  $\text{colPic}$ .

- De lo contrario ( $(\text{yP} \gg \text{Log2CtbSize})$  no es igual a  $(\text{yPRb} \gg \text{Log2CtbSize})$ ),  $\text{colPu}$  se marca como "no disponible".

2. Cuando  $\text{colPu}$  se codifica en un modo de intrapredicción o  $\text{colPu}$  se marca como "no disponible", se aplica lo siguiente.

- La posición de luma central de la unidad de predicción actual se define mediante

$$\text{xPCtr} = (\text{xP} + \text{nPSW} \gg 1) \quad (8-141)$$

$$\text{yPCtr} = (\text{yP} + \text{nPSH} \gg 1) \quad (8-142)$$

- La variable  $\text{colPu}$  se establece como la unidad de predicción que cubre la posición modificada dada por  $((\text{xPCtr} \gg 4) \ll 4, (\text{yPCtr} \gg 4) \ll 4)$  dentro de  $\text{colPic}$ .

3. ( $\text{xPCol}$ ,  $\text{yPCol}$ ) se establece igual a la muestra de luma superior izquierda de  $\text{colPu}$  en relación con la muestra de luma superior izquierda de  $\text{colPic}$ .

La función  $\text{LongTermRefPic}(\text{picX}, \text{refIdx}, \text{LX})$  se puede definir como sigue: si la imagen de referencia con índice  $\text{refIdx}$  de la lista de imágenes de referencia LX de la imagen  $\text{picX}$  se marcó como "usada como referencia a largo plazo" en el momento en que  $\text{picX}$  era la imagen actual,  $\text{LongTermRefPic}(\text{picX}, \text{refIdx}, \text{LX})$  devuelve 1; de lo contrario,  $\text{LongTermRefPic}(\text{picX}, \text{refIdx}, \text{LX})$  devuelve 0.

El codificador de vídeo 20 y el descodificador de vídeo 30 pueden implementar una versión modificada de la función "AddPicID()" de HEVC. Por ejemplo, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden implementar  $\text{AddPicID}(\text{picX}, \text{refIdx}, \text{LX})$  de modo que esta función devuelva  $\text{AddPicID}(\text{pic})$ , donde "pic" es la imagen de referencia con índice  $\text{refIdx}$  de la lista de imágenes de referencia LX de la imagen  $\text{picX}$ .

El codificador de vídeo 20 y el descodificador de vídeo 30 pueden obtener las variables  $\text{mvLXCol}$  y  $\text{availableFlagLXCol}$  como sigue, donde el texto subrayado representa cambios con respecto a

HEVC WD7:

- Si se cumple una o más de las siguientes condiciones, ambas componentes de mvLXCol se establecen igual a 0 y availableFlagLXCol se establece igual a 0.

- colPu se codifica en un modo de intrapredicción.

- colPu se marca como "no disponible".

- pic\_temporal\_mvp\_enable\_flag es igual a 0.

- De lo contrario, el vector de movimiento mvCol, el índice de referencia refIdxCol y el identificador de lista de referencia listCol se obtienen como sigue.

- Si PredFlagL0[xPCol][yPCol] es igual a 0, mvCol, refIdxCol y listCol se establecen igual a MvL1[xPCol][yPCol], RefIdxL1[xPCol][yPCol] y L1, respectivamente.

- De lo contrario (PredFlagL0[xPCol][yPCol] es igual a 1), se aplica lo siguiente.

- Si PredFlagL1[xPCol][yPCol] es igual a 0, mvCol, refIdxCol y listCol se establecen igual a MvL0[xPCol][yPCol], RefIdxL0[xPCol][yPCol] y L0, respectivamente.

- De lo contrario (PredFlagL1[xPCol][yPCol] es igual a 1), se realizan las siguientes asignaciones.

- Si PicOrderCnt(pic) de cada imagen pic en cada lista de imágenes de referencia es menor que o igual a PicOrderCntVal, mvCol, refIdxCol y listCol se establecen igual a MvLX[xPCol][yPCol], RefIdxLX[xPCol][yPCol] y LX, respectivamente, siendo X el valor de X para el que se invoca este proceso.

- De lo contrario (PicOrderCnt(pic) de al menos una imagen pic en al menos una lista de imágenes de referencia es mayor que PicOrderCntVal), mvCol, refIdxCol y listCol se establecen igual a MvLN[xPCol][yPCol], RefIdxLN[xPCol][yPCol] y LN, respectivamente, siendo N el valor de collocated\_from\_l0\_flag.

- Si se cumple una de las siguientes condiciones, la variable availableFlagLXCol se establece igual a 0.

- AddPicId(RefPicListX[refIdxLX]) no es igual a AddPicId(colPic, refIdxCol, listCol);

- RefPicListX[refIdxLX] es una imagen de referencia a corto plazo y LongTermRefPic(colPic, refIdxCol, listCol) es igual a 1;

- RefPicListX[refIdxLX] es una imagen de referencia a largo plazo y LongTermRefPic(colPic, refIdxCol, listCol) es igual a 0;

- De lo contrario, la variable availableFlagLXCol se establece igual a 1 y se aplica lo siguiente.

- Si RefPicListX[refIdxLX] es una imagen de referencia a largo plazo, o LongTermRefPic(colPic, refIdxCol, listCol) es igual a 1, o PicOrderCnt(colPic) - RefPicOrderCnt(colPic, refIdxCol, listCol) es igual a PicOrderCntVal - PicOrderCnt(RefPicListX[refIdxLX]),

$$mvLXCol = mvCol \quad (8-143)$$

- De lo contrario, mvLXCol se obtiene como una versión escalada del vector de movimiento mvCol como se especifica a continuación.

$$tx = (16384 + (Abs(td) \gg 1)) / td \quad (8-144)$$

$$DistScaleFactor = Clip3(-4096, 4095, (tb * tx + 32) \gg 6) \quad (8-145)$$

$$mvLXCol =$$

$$Clip3(-8192, 8191.75, Sign(DistScaleFactor * mvCol) * ((Abs(DistScaleFactor * mvCol) + 127) \gg 8)) \quad (8-146)$$

donde td y tb pueden obtenerse como

$$td = Clip3(-128, 127, PicOrderCntVal(colPic) - RefPicOrderCnt(colPic, refIdxCol, listCol)) \quad (8-147)$$

$$td = Clip3(-128, 127, PicOrderCntVal - PicOrderCnt(RefPicListX[refIdxLX])) \quad (8-148)$$

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para realizar un proceso de obtención para candidatos combinados de fusión bipredictiva. Los datos de entrada de este proceso pueden incluir una lista de candidatos de fusión mergeCandList, índices de referencia refIdxL0N y refIdxL1N de cada candidato N que está en mergeCandList, banderas de utilización de lista de predicción predFlagL0N y predFlagL1N de cada candidato N que está en mergeCandList, vectores de movimiento mvL0N y mvL1N de cada candidato N que está en mergeCandList, el número de elementos numMergeCand dentro de mergeCandList y el número de elementos numOrigMergeCand dentro de mergeCandList después del proceso de obtención de candidato de fusión espacial y temporal. Los datos de salida de este proceso pueden incluir la lista de candidatos de fusión mergeCandList, el número de elementos numMergeCand dentro de mergeCandList, índices de referencia refIdxL0combCandk y refIdxL1combCandk de cada nuevo candidato combCandk que se añade en mergeCandList durante la invocación de este proceso, banderas de utilización de lista de predicción predFlagL0combCandk y predFlagL1combCandk de cada nuevo candidato combCandk que se añade en mergeCandList durante la invocación de este proceso, y vectores de movimiento mvL0combCandk y mvL1combCandk de cada nuevo candidato combCandk que se añaden en mergeCandList durante la invocación de este proceso.

Este proceso se puede definir como sigue, donde el texto subrayado representa cambios con respecto a HEVC WD7 y las elipsis representan texto que es el mismo que HEVC WD7. Cuando numOrigMergeCand es mayor que 1 y menor que MaxNumMergeCand, la variable numInputMergeCand puede establecerse como numMergeCand, las variables combIdx y combCnt pueden establecerse en 0, la variable combStop puede establecerse en FALSE y las siguientes etapas pueden repetirse hasta que combStop sea igual a TRUE:

1. Las variables l0CandIdx e l1CandIdx se obtienen usando combIdx como se especifica en la Tabla 8-8.

2. Las siguientes asignaciones se realizan con l0Cand como candidato en la posición l0CandIdx e l1Cand como el candidato en la posición l1CandIdx en la lista de candidatos de fusión mergeCandList (l0Cand = mergeCandList[l0CandIdx], l1Cand = mergeCandList[l1CandIdx]).

3. Cuando se cumplen todas las condiciones siguientes,

- predFlagL0l0Cand = 1

- predFlagL1l1Cand = 1

-  $\frac{\text{AddPicId}(\text{RefPicListL0}[\text{refIdxL0l0Cand}])}{\text{PicOrderCnt}(\text{RefPicList0}[\text{refIdxL0l0Cand}])} \neq \frac{\text{AddPicId}(\text{RefPicListL1}[\text{refIdxL1l1Cand}])}{\text{PicOrderCnt}(\text{RefPicList1}[\text{refIdxL1l1Cand}])} \mid \text{mvL0l0Cand} \neq \text{mvL1l1Cand}$  se aplica lo siguiente.

4. ...

5. ...

En algunos ejemplos, la predicción entre dos vectores de movimiento que hacen referencia a dos imágenes de referencia a largo plazo diferentes puede estar inhabilitada. En otros ejemplos, la predicción entre dos vectores de movimiento que hacen referencia a dos imágenes de referencia diferentes entre vistas puede estar inhabilitada.

De esta manera, el codificador de vídeo 20 y el decodificador de vídeo 30 representan ejemplos de un codificador de vídeo configurado para codificar un valor de recuento de orden de imagen (POC) para una primera imagen de datos de vídeo, codificar un identificador de imagen de segunda dimensión para la primera imagen y codificar una segunda imagen en base a, al menos en parte, el valor de POC y el identificador de imagen de segunda dimensión de la primera imagen. La codificación de la segunda imagen en base al valor de POC y el identificador de imagen de segunda dimensión de la primera imagen puede incluir identificar la primera imagen usando tanto el valor de POC de la primera imagen como el identificador de imagen de segunda dimensión.

Además, como se muestra anteriormente, la codificación de la segunda imagen puede incluir habilitar o inhabilitar la predicción de vectores de movimiento para un vector de movimiento que hace referencia a la primera imagen, en base al valor de POC y el identificador de imagen de segunda dimensión de la primera imagen y un valor de POC y un identificador de imagen de segunda dimensión de una imagen de referencia a la que hace referencia un predictor de vector de movimiento candidato. De acuerdo con la invención, si el identificador de imagen de segunda dimensión de la primera imagen indica que la primera imagen es una imagen a corto plazo, y el identificador de imagen de segunda dimensión de la imagen de referencia indica que la imagen de referencia es una imagen de referencia a largo plazo, el codificador de vídeo 20 y el decodificador de vídeo 30 inhabilitan la predicción de vectores de movimiento entre un vector de movimiento que hace referencia a la primera imagen y un vector de movimiento que hace referencia a la imagen de referencia.

Además, como también se muestra anteriormente, codificar la segunda imagen puede incluir codificar un vector de movimiento de un bloque de la segunda imagen que hace referencia a la primera imagen, como se indica anteriormente. Dicha codificación puede basarse en el valor de POC de la primera imagen en el sentido de que, si un predictor de vector de movimiento hace referencia a una imagen de referencia que tiene un valor de POC diferente, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden configurarse para escalar el predictor de vector de movimiento en base a diferencias de valor de POC entre la primera y la segunda imagen, y la imagen de referencia y la segunda imagen.

El codificador de vídeo 20 y el descodificador de vídeo 30 pueden configurarse para realizar las técnicas de cualquiera o todos los ejemplos descritos anteriormente, solos o en cualquier combinación. Tanto el codificador de vídeo 20 como el descodificador de vídeo 30 se pueden implementar como cualquiera de una variedad de circuitos codificadores o descodificadores adecuados, según corresponda, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), matrices de puertas programables *in situ* (FPGA), circuitos de lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. Tanto el codificador de vídeo 20 como el descodificador de vídeo 30 pueden estar incluidos en uno o más codificadores o descodificadores, cualquiera de los cuales puede estar integrado como parte de un codificador/descodificador (CÓDEC) de vídeo combinado. Un dispositivo que incluye un codificador de vídeo 20 y/o un descodificador de vídeo 30 puede comprender un circuito integrado, un microprocesador y/o un dispositivo de comunicación inalámbrica, tal como un teléfono celular o un ordenador de tipo tableta.

La FIG. 2 es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo 20 que puede implementar técnicas para codificar datos de vídeo de acuerdo con una ampliación solamente de sintaxis de alto nivel de una norma de codificación de vídeo. El codificador de vídeo 20 puede realizar intracodificación e intercodificación de bloques de vídeo dentro de segmentos de vídeo. La intracodificación está basada en la predicción espacial para reducir o eliminar la redundancia espacial en vídeo dentro de una trama o imagen de vídeo dada. La intercodificación está basada en la predicción temporal para reducir o eliminar la redundancia temporal en vídeo dentro de tramas o imágenes contiguas de una secuencia de vídeo. El modo 'intra' (modo I) se puede referir a cualquiera de varios modos de codificación espacial. Los modos 'inter', tal como la predicción unidireccional (modo P) o la bipredicción (modo B), se pueden referir a cualquiera de varios modos de codificación 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 a codificar. En el ejemplo de la FIG. 2, el codificador de vídeo 20 incluye 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. La unidad de selección de modo 40 incluye, a su vez, 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 límites de bloques para eliminar las distorsiones de efecto pixelado del vídeo reconstruido. Si se desea, el filtro de eliminación de bloques podría filtrar típicamente la salida del sumador 62. También se pueden usar filtros adicionales (en bucle o posbucle) además del filtro de eliminación de bloques. Dichos filtros no se muestran para una mayor 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 segmento de vídeo que se ha de codificar. La trama o el segmento se puede dividir en múltiples bloques de vídeo. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 realizan una codificación interpredictiva del bloque de vídeo recibido, en relación con uno o más bloques de una o más tramas de referencia para proporcionar una predicción temporal. La unidad de compensación de movimiento 44 puede codificar un vector de movimiento de acuerdo con las técnicas de esta divulgación, por ejemplo, durante la predicción avanzada de vectores de movimiento (AMVP), la predicción temporal de vectores de movimiento (TMVP) o la codificación en modo de fusión. De forma alternativa, la unidad de intrapredicción 46 puede realizar una codificación intrapredictiva del bloque de vídeo recibido, en relación con uno o más bloques vecinos en la misma trama o segmento que el bloque a codificar, para proporcionar predicción espacial. El codificador de vídeo 20 puede realizar múltiples pasadas de codificación, por ejemplo, para seleccionar un modo de codificación apropiado para cada bloque de datos de vídeo.

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

La unidad de selección de modo 40 puede seleccionar uno de los modos de codificación, intra o inter, por ejemplo, en base a resultados erróneos, y proporciona el bloque intracodificado o intercodificado resultante al sumador 50 para generar datos de bloque 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 modo "intra", 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 generar vectores de movimiento, que estima 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) con respecto al bloque actual que se está codificando dentro de la trama actual (u otra unidad codificada). Un bloque predictivo es un bloque que coincide estrechamente con el bloque que se va a codificar, en términos de diferencia de píxeles, que se puede determinar mediante una suma de diferencia absoluta (SAD), una suma de diferencia al cuadrado (SSD) u otras métricas de diferencia. 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 relativa a las posiciones de píxel completo y las posiciones de píxel fraccionario, y proporcionar 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 segmento 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 entre una primera lista de imágenes de referencia (lista 0) o una segunda lista de imágenes de referencia (lista 1), cada una de las cuales identifica una o más imágenes de referencia almacenadas en la memoria de imágenes de referencia 64. La unidad de estimación de movimiento 42 envía el vector de movimiento calculado a la unidad de codificación 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 obtener 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 pueden estar funcionalmente integradas, en algunos ejemplos. Al recibir el vector de movimiento para la PU del bloque de vídeo actual, la unidad de compensación de movimiento 44 puede localizar el bloque predictivo al que apunta el vector de movimiento en una de las listas de imágenes de referencia. El 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 diferencia de píxeles, como se analiza a continuación. En general, la unidad de estimación de movimiento 42 realiza una estimación de movimiento en relación con componentes de luma, y la unidad de compensación de movimiento 44 usa vectores de movimiento calculados en base a las componentes de luma, tanto para componentes de croma como para componentes de luma. La unidad de selección de modo 40 también puede generar elementos de sintaxis asociados a los bloques de vídeo y al segmento de vídeo, para su uso por el decodificador de vídeo 30 en la decodificación de los bloques de vídeo del segmento de vídeo.

La unidad de intrapredicción 46 puede intrapredecir un bloque actual, como 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 que se va a 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 diferentes pasadas de codificación, y la unidad de intrapredicción 46 (o la unidad de selección de modo 40, en algunos ejemplos) puede seleccionar de entre los modos probados un modo de intrapredicción adecuado que se va a usar.

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 probados, y seleccionar el modo de intrapredicción que tiene las mejores características de velocidad-distorsión entre los modos probados. El análisis de velocidad-distorsión determina, en general, una cantidad de distorsión (o error) entre un bloque codificado y un bloque original no codificado que se ha codificado para generar el bloque codificado, así como una velocidad de bits (es decir, un número de bits) usada para generar el bloque codificado. La unidad de intrapredicción 46 puede calcular relaciones 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 a la unidad de codificación entrópica 56 información indicativa del modo de intrapredicción seleccionado para el bloque. 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 modos de intrapredicción y una pluralidad de tablas de índices de modos de intrapredicción modificados (también denominadas tablas de mapeo de palabras de código), definiciones de contextos de codificación para diversos bloques, e indicaciones del modo de intrapredicción



más probable, una tabla de índices de modos de intrapredicción y una tabla de índices de modos de intrapredicción modificados que se va a usar para cada uno de los contextos.

El codificador de vídeo 20 forma un bloque de vídeo residual restando los datos de predicción de la unidad de selección de modo 40 al bloque de vídeo original que se está codificando. El sumador 50 representa el componente o componentes que realizan esta operación de resta. La unidad de procesamiento de transformada 52 aplica una transformada, tal como una transformada de coseno discreta (DCT) o una transformada conceptualmente similar, al bloque residual, generando 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, generando un bloque de coeficientes de transformada residuales. La transformada puede convertir la información residual de un dominio de valor 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 aún más la velocidad de bits. El proceso de cuantificación puede reducir la profundidad de bits asociada a 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 entrópicamente los coeficientes de transformada cuantificados. Por ejemplo, la unidad de codificación entrópica 56 puede realizar una codificación de longitud variable adaptativa al contexto (CAVLC), codificación aritmética binaria adaptativa al contexto (CABAC), codificación aritmética binaria adaptativa al contexto basada en sintaxis (SBAC), codificación 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 puede estar basado en bloques vecinos. Después de la codificación entrópica mediante 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 una cuantificación inversa y una transformada inversa, respectivamente, para reconstruir el bloque residual en el dominio de píxel, por ejemplo, para su uso posterior como un bloque de referencia. La unidad de compensación de movimiento 44 puede calcular un bloque de referencia sumando 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 píxel fraccionario, para su uso en la estimación de movimiento. El sumador 62 suma el bloque residual reconstruido al bloque de predicción sometido a compensación de movimiento generado por la unidad de compensación de movimiento 44 para generar un bloque de vídeo reconstruido para su almacenamiento en la memoria de imágenes de referencia 64. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden usar el bloque de vídeo reconstruido como bloque de referencia para intercodificar un bloque en una trama de vídeo subsiguiente.

El codificador de vídeo 20 puede configurarse para realizar cualquiera o todas las diversas técnicas de ejemplo analizadas con respecto a la FIG. 1, por sí solas o en cualquier combinación. Por ejemplo, de acuerdo con las técnicas de esta divulgación, el codificador de vídeo 20 puede codificar una imagen basándose en un valor de recuento de orden de imagen (POC) de una imagen de referencia y un identificador de segunda dimensión de la imagen de referencia. Es decir, el codificador de vídeo 20 puede codificar un valor de POC de una primera imagen (una imagen de referencia, en este ejemplo), así como un identificador de imagen de segunda dimensión para la primera imagen. El identificador de imagen de segunda dimensión puede comprender, por ejemplo, un identificador de vista para una vista que incluye la primera imagen, un índice de orden de vista para la vista que incluye la primera imagen, una combinación del índice de orden de vista y una bandera de profundidad, un identificador de capa para una capa de codificación de vídeo escalable (SVC) que incluye la primera imagen, y un identificador de capa genérico.

El identificador de segunda dimensión puede comprender, de forma adicional o alternativa, un valor que indica si la primera imagen es una imagen de referencia a largo plazo o una imagen de referencia a corto plazo. De forma alternativa, no de acuerdo con la invención, un valor aparte puede indicar si la primera imagen es una imagen de referencia a corto o largo plazo, además del valor de POC y el identificador de imagen de segunda dimensión. En algunos ejemplos, las indicaciones de a largo y corto plazo para imágenes de referencia pueden indicar si las imágenes de referencia son imágenes de referencia temporales o imágenes de referencia entre vistas. Por ejemplo, una imagen de referencia a largo plazo puede corresponder a una imagen de referencia temporal (es decir, una imagen de referencia en la misma capa o vista), mientras que una imagen de referencia a corto plazo puede corresponder a una imagen de referencia entre vistas. Como otro ejemplo, una imagen de referencia a largo plazo puede corresponder a una imagen de referencia entre vistas, mientras que una imagen de referencia a corto plazo puede corresponder a una imagen de referencia temporal.

Asimismo, el codificador de vídeo 20 puede inhabilitar la predicción de vectores de movimiento entre vectores de movimiento de diferentes tipos. Los "tipos" de vectores de movimiento pueden incluir, por ejemplo, vectores de movimiento temporal, que hacen referencia a imágenes de referencia temporal (es decir, imágenes en la misma vista que una imagen actual que se está codificando), y vectores de movimiento de disparidad, que hacen referencia a imágenes de referencia entre vistas (es decir, imágenes en una vista diferente a la vista que incluye la imagen actual). Típicamente, las imágenes de referencia entre vistas tienen el mismo valor de POC que la imagen actual. Es decir, típicamente, las imágenes de referencia entre vistas y la imagen actual se producen en la misma unidad de acceso. El codificador de vídeo 20 puede inhabilitar la predicción de vectores de movimiento entre vectores de movimiento de diferentes tipos. Es decir, si un vector de movimiento actual de la imagen actual es un vector de movimiento temporal, el codificador de vídeo 20 puede inhabilitar la predicción de vectores de movimiento con respecto a un vector de movimiento de disparidad. Asimismo, si el vector de movimiento actual es un vector de movimiento de disparidad, el codificador de vídeo 20 puede inhabilitar la predicción de vectores de movimiento con respecto a un vector de movimiento temporal. De lo contrario, el codificador de vídeo 20 puede codificar el vector de movimiento actual usando un proceso de codificación de vectores de movimiento, tal como la predicción avanzada de vectores de movimiento (AMVP) o el modo de fusión.

En algunos ejemplos, el codificador de vídeo 20 puede configurarse para codificar un valor que indica si la primera imagen (por ejemplo, una componente de vista, en la codificación de vídeo multivista) es una imagen de referencia a largo plazo basada, al menos en parte, en si se usa la primera imagen para la predicción entre vistas. Por ejemplo, el codificador de vídeo 20 puede codificar un elemento de sintaxis que indica si la primera imagen es una imagen de referencia a corto o largo plazo en un conjunto de parámetros de secuencia (SPS) correspondiente a una secuencia que incluye la primera imagen.

El codificador de vídeo 20 puede configurarse para marcar imágenes de referencia entre vistas como imágenes de referencia a largo plazo, al menos temporalmente. El codificador de vídeo 20 puede almacenar además los estados actuales de las imágenes de referencia entre vistas, donde los estados pueden comprender una imagen de referencia a largo plazo, una imagen de referencia a corto plazo y no usada como referencia. Por tanto, si la primera imagen comprende una imagen entre vistas, el codificador de vídeo 20 puede marcar la primera imagen como una imagen de referencia a largo plazo. Después de codificar una segunda imagen con relación a la primera imagen, el codificador de vídeo 20 puede restaurar un estado para la imagen de referencia entre vistas basándose en el estado almacenado.

El codificador de vídeo 20 puede asignar temporalmente nuevos valores de POC a imágenes de referencia entre vistas mientras codifica la segunda imagen. El codificador de vídeo 20 puede determinar un conjunto de valores de POC para las imágenes de referencia temporales actuales y asignar valores de POC no usados a las imágenes de referencia entre vistas. El codificador de vídeo 20 también puede almacenar respectivos valores actuales de POC para cada imagen de referencia entre vistas. Después de codificar la segunda imagen, el codificador de vídeo 20 puede restaurar los valores de POC almacenados (es decir, originales) para las imágenes de referencia entre vistas. Debido a que las imágenes de referencia entre vistas están típicamente en la misma unidad de acceso que la segunda imagen (es decir, la imagen que se está codificando actualmente), el codificador de vídeo 20 puede, en cambio, simplemente establecer los valores de POC para las imágenes de referencia entre vistas iguales al valor de POC de la segunda imagen, es decir, la imagen actual que se está codificando, de modo que no es necesario almacenar los valores de POC.

De esta manera, el codificador de vídeo 20 de la FIG. 2 representa un ejemplo de un codificador de vídeo configurado para codificar un valor de recuento de orden de imagen (POC) para una primera imagen de datos de vídeo, codificar un identificador de imagen de segunda dimensión para la primera imagen y codificar, de acuerdo con una especificación base de codificación de vídeo, una segunda imagen en base a, al menos en parte, el valor de POC y el identificador de imagen de segunda dimensión de la primera imagen. La especificación base de codificación de vídeo puede comprender HEVC. Además, el codificador de vídeo 20 puede configurarse para codificar una imagen de acuerdo con una ampliación de la especificación base de codificación de vídeo, por ejemplo, una ampliación SVC o MVC de HEVC. Por tanto, el codificador de vídeo 20 representa un ejemplo de un codificador de vídeo configurado para codificar un valor de recuento de orden de imagen (POC) para una primera imagen de datos de vídeo, codificar un identificador de imagen de segunda dimensión para la primera imagen y codificar, de acuerdo con una ampliación de una especificación base de codificación de vídeo, una segunda imagen en base a, al menos en parte, el valor de POC y el identificador de imagen de segunda dimensión de la primera imagen.

La FIG. 3 es un diagrama de bloques que ilustra un ejemplo de un descodificador de vídeo 30 que puede implementar técnicas para codificar datos de vídeo de acuerdo con una ampliación solamente de sintaxis de alto nivel de una norma de codificación de vídeo. En el ejemplo de la FIG. 3, el descodificador de vídeo 30 incluye una unidad de descodificación entrópica 70, una unidad de compensación de movimiento 72, una unidad de intrapredicción 74, una unidad de cuantificación inversa 76, una unidad de transformada inversa 78, una memoria de imágenes de referencia 82 y un sumador 80. El descodificador de vídeo 30 puede, en algunos ejemplos, realizar una pasada de descodificación generalmente recíproca a la pasada de codificación descrita con respecto al codificador de vídeo 20 (FIG. 2). La unidad de compensación de movimiento 72 puede generar datos de predicción 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 segmento de vídeo codificado y elementos de sintaxis asociados desde el codificador de vídeo 20. La unidad de descodificación entrópica 70 del descodificador de vídeo 30 descodifica entrópicamente 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 descodificación entrópica 70 reenvía los vectores de movimiento y otros elementos de sintaxis a la unidad de compensación de movimiento 72. El descodificador de vídeo 30 puede recibir los elementos de sintaxis a nivel de segmento de vídeo y/o a nivel de bloque de vídeo.

Cuando el segmento de vídeo se codifica como un segmento intracodificado (I), la unidad de intrapredicción 74 puede generar datos de predicción para un bloque de vídeo del segmento de vídeo actual, basándose en un modo de intrapredicción señalado y en datos de bloques descodificados previamente de la trama o imagen actual. Cuando la trama de vídeo se codifica como un segmento intercodificado (es decir, B, P o GPB), la unidad de compensación de movimiento 72 genera bloques predictivos para un bloque de vídeo del segmento de vídeo actual en base a los vectores de movimiento y otros elementos de sintaxis recibidos desde la unidad de descodificación entrópica 70. Los bloques predictivos se pueden generar a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El descodificador de vídeo 30 puede generar las listas de tramas de referencia, Lista 0 y Lista 1, usando técnicas de generación predeterminadas en base a imágenes de referencia almacenadas en la memoria de imágenes de referencia 82.

De acuerdo con las técnicas de esta divulgación, la unidad de descodificación entrópica 70 puede descodificar datos codificados entrópicamente representativos de información de movimiento para un bloque actual de una imagen actual. Por ejemplo, de acuerdo con AMVP, la unidad de descodificación entrópica 70 puede descodificar valores de diferencia de vector de movimiento (MVD) para el bloque actual. La unidad de compensación de movimiento 72 (u otra unidad del descodificador de vídeo 30, tal como la unidad de descodificación entrópica 70) puede reconstruir el vector de movimiento para el bloque actual usando la información de movimiento descodificada entrópicamente, tal como los valores de MVD. Por ejemplo, la unidad de compensación de movimiento 72 puede determinar un conjunto de predictores de vector de movimiento disponibles para el vector de movimiento actual, por ejemplo, basándose en si el vector de movimiento actual hace referencia a una imagen de referencia a largo plazo o una imagen de referencia a corto plazo (o una imagen de referencia temporal o entre vistas), y en si un conjunto de imágenes de referencia candidatas también hace referencia a imágenes de referencia a corto o largo plazo (o imágenes de referencia temporal o entre vistas).

Como se analizó anteriormente, la unidad de compensación de movimiento 72 puede determinar que los predictores de vector de movimiento candidatos de diferentes tipos no están disponibles para su uso para predecir un vector de movimiento actual. Por ejemplo, de una manera que no pertenece a la invención, cuando el vector de movimiento actual es un vector de movimiento temporal, la unidad de compensación de movimiento 72 puede determinar que los vectores de movimiento de disparidad no están disponibles para su uso como predictores de vector de movimiento para el vector de movimiento actual. Asimismo, cuando el vector de movimiento actual es un vector de movimiento de disparidad, la unidad de compensación de movimiento 72 puede determinar que los vectores de movimiento temporal no están disponibles para su uso como predictores de vector de movimiento para el vector de movimiento actual. De acuerdo con la invención, la unidad de compensación de movimiento 72 también inhabilita la predicción de vectores de movimiento entre imágenes de referencia a largo y corto plazo.

En caso de que el vector de movimiento actual sea un vector de movimiento de disparidad, la unidad de compensación de movimiento 72 también puede evitar escalar un predictor de vector de movimiento (que puede, igualmente, corresponder a un vector de movimiento de disparidad). Además, o como la alternativa no pertenece a la invención, la unidad de compensación de movimiento 72 puede asignar un valor de POC temporal a una imagen de referencia entre vistas a la que hace referencia un predictor de vector de movimiento de disparidad durante la predicción de vectores de movimiento de un vector de movimiento de disparidad.

En cualquier caso, la unidad de compensación de movimiento 72, u otro elemento del descodificador de vídeo 30, puede reproducir un vector de movimiento para un bloque actual, por ejemplo, usando AMVP o el modo de fusión. La unidad de compensación de movimiento 72 determina información de predicción para un bloque de vídeo del segmento de vídeo actual, analizando sintácticamente los vectores de movimiento y otros elementos de sintaxis, y usa la información de predicción para generar los bloques predictivos para el bloque de vídeo actual que se está descodificando. Por ejemplo, la unidad de compensación de movimiento 72 usa algunos de los elementos de sintaxis recibidos para determinar un modo de predicción (por ejemplo, intrapredicción o interpredicción) usado para codificar los bloques de vídeo del segmento de vídeo, un tipo de segmento de interpredicción (por ejemplo, segmento B, segmento P o segmento GPB), información de generación para una o más de las listas de imágenes de referencia para el segmento, vectores de movimiento para cada bloque de vídeo intercodificado del segmento, el estado de interpredicción para cada bloque de vídeo intercodificado del segmento y otra información para descodificar los bloques de vídeo en el segmento de vídeo actual. La unidad de compensación de movimiento 72 puede codificar un vector de movimiento de acuerdo con las técnicas de esta divulgación, por ejemplo, durante la predicción avanzada de vectores de movimiento (AMVP), la predicción temporal de vectores de movimiento (TMVP) o la codificación en modo de fusión.

La unidad de compensación de movimiento 72 también puede realizar una interpolación en base a filtros de interpolación. La unidad de compensación de movimiento 72 puede usar filtros de interpolación como los que usa 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 elementos de sintaxis recibidos y usar los filtros de interpolación para generar bloques predictivos.

La unidad de cuantificación inversa 76 realiza la cuantificación inversa, es decir, la descuantificación, de los coeficientes de transformada cuantificados proporcionados en el flujo de bits y descodificados por la unidad de descodificación entrópica 70. El proceso de cuantificación inversa puede incluir el uso de un parámetro de cuantificación QP<sub>y</sub> calculado por el descodificador de vídeo 30 para cada bloque de vídeo del segmento de vídeo para determinar un grado de cuantificación y, de igual modo, un grado de cuantificación inversa que se debería aplicar. La unidad de transformada inversa 78 aplica una transformada inversa, por ejemplo, una DCT inversa, una transformada de enteros inversa o un proceso de transformada inversa conceptualmente similar, a los coeficientes de transformada para generar bloques residuales en el dominio de píxel.

Después de que la unidad de compensación de movimiento 72 genere 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 con los correspondientes bloques predictivos generados por la unidad de compensación de movimiento 72. El sumador 80 representa el componente o componentes que realizan esta operación de suma. Si se desea, también se puede aplicar un filtro de eliminación de bloques para filtrar los bloques descodificados para eliminar las distorsiones de efecto pixelado. También se pueden usar otros filtros de bucle (ya sea en el bucle de codificación o después del bucle de codificación) para suavizar las transiciones entre píxeles o mejorar de otro modo la calidad del vídeo. Los bloques de vídeo descodificados en una trama o imagen dadas se almacenan a continuación en la memoria de imágenes de referencia 82, que almacena imágenes de referencia usadas para una compensación de movimiento subsiguiente. La memoria de imágenes de referencia 82 también almacena vídeo descodificado para una presentación posterior en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la FIG. 1.

De esta manera, el descodificador de vídeo 30 de la FIG. 3 representa un ejemplo de un descodificador de vídeo configurado para descodificar un valor de recuento de orden de imagen (POC) para una primera imagen de datos de vídeo, descodificar un identificador de imagen de segunda dimensión para la primera imagen y descodificar una segunda imagen en base a, al menos en parte, el valor de POC y el identificador de imagen de segunda dimensión de la primera imagen. La especificación base de codificación de vídeo puede comprender HEVC. Además, el descodificador de vídeo 30 puede configurarse para codificar una imagen de acuerdo con una ampliación de la especificación base de codificación de vídeo, por ejemplo, una ampliación SVC o MVC de HEVC. De esta manera, el descodificador de vídeo 30 también representa un ejemplo de un descodificador de vídeo configurado para descodificar un valor de recuento de orden de imagen (POC) para una primera imagen de datos de vídeo, descodificar un identificador de imagen de segunda dimensión para la primera imagen y descodificar, de acuerdo con una ampliación de una especificación base de codificación de vídeo, una segunda imagen en base a, al menos en parte, el valor de POC y el identificador de imagen de segunda dimensión de la primera imagen.

La FIG. 4 es un diagrama conceptual que ilustra un ejemplo de patrón de predicción MVC. La codificación de vídeo multivista (MVC) es una ampliación de ITU-T H.264/AVC. Se puede aplicar una técnica similar a HEVC. En el ejemplo de la FIG. 4, se ilustran ocho vistas (que tienen ID de vista "S0" a "S7"), y se ilustran doce ubicaciones temporales ("T0" a "T11") para cada vista. Es decir, cada fila de la FIG. 4 corresponde a una vista, mientras que cada columna indica una ubicación temporal.

Aunque MVC tiene una denominada vista base que puede descodificarse por descodificadores H.264/AVC y un par de vistas estéreo también podría ser compatible con MVC, una ventaja de MVC es que podría admitir un ejemplo que usa más de dos vistas como datos de entrada de vídeo 3D y descodifica este vídeo 3D representado por las múltiples vistas. Un renderizador de un cliente que tiene un descodificador MVC puede esperar contenido de vídeo 3D con múltiples vistas.

Una disposición típica de orden de descodificación MVC se conoce como codificación de tiempo primero. Una unidad de acceso puede incluir imágenes codificadas de todas las vistas para una instancia de tiempo de salida. Por ejemplo, cada una de las imágenes de tiempo T0 puede incluirse en una unidad de acceso común, cada una de las imágenes de tiempo T1 puede estar incluida en una segunda unidad de acceso común, y así sucesivamente. El orden de descodificación no es necesariamente idéntico al orden de salida o visualización.

Las tramas, es decir, imágenes, en la FIG. 4 se indican en la intersección de cada fila y cada columna en la FIG. 4 usando un bloque sombreado que incluye una letra, que indica si la trama correspondiente está intracodificada (es decir, una trama I) o intercodificada en una dirección (es decir, como una trama P) o en múltiples direcciones (es decir, como trama B). En general, las predicciones se indican mediante flechas, donde la trama apuntada usa como

referencia de predicción el objeto desde el que se apunta . Por ejemplo, la trama P de la vista S2 en la ubicación temporal T0 se predice a partir de la trama I de la vista SO en la ubicación temporal T0.

Al igual que con la codificación de vídeo de una sola vista, las tramas de una secuencia de vídeo de codificación de vídeo multivista pueden codificarse de forma predictiva con respecto a tramas en diferentes ubicaciones temporales. Por ejemplo, la trama b de la vista SO en la ubicación temporal T1 tiene una flecha que apunta a la misma desde la trama I de la vista SO en la ubicación temporal T0, lo que indica que la trama b se predice a partir de la trama I. Además, sin embargo, en el contexto de la codificación de vídeo multivista, las tramas pueden predecirse entre vistas. Es decir, una componente de vista puede usar los componentes de vista en otras vistas como referencia. En MVC, por ejemplo, la predicción entre vistas se realiza como si la componente de vista en otra vista fuera una referencia entre predicciones. Las posibles referencias entre vistas se indican en la ampliación MVC del conjunto de parámetros de secuencia (SPS) y se pueden modificar mediante el proceso de generación de lista de imágenes de referencia, lo que permite una ordenación flexible de las referencias de predicción entre vistas o entre predicciones.

En la ampliación MVC de H.264/AVC, como ejemplo, la predicción entre vistas es compatible con la compensación de movimiento de disparidad, que usa la sintaxis de la compensación de movimiento H.264/AVC, pero permite que una imagen en una vista diferente se use como imagen de referencia. La codificación de dos vistas puede ser compatible con MVC, que generalmente se conoce como vistas estereoscópicas. Una de las ventajas de MVC es que un codificador MVC podría tomar más de dos vistas como datos de entrada de vídeo 3D y un descodificador MVC puede decodificar dicha representación multivista. Por lo tanto, un dispositivo de renderizado con un descodificador MVC puede esperar contenidos de vídeo 3D con más de dos vistas.

En MVC, se permite la predicción entre vistas (IVP) entre imágenes en la misma unidad de acceso (es decir, con la misma instancia de tiempo). Una unidad de acceso es, en general, una unidad de datos que incluye todos los componentes de vista (por ejemplo, todas las unidades NAL) para una instancia temporal común. Por lo tanto, en MVC, se permite la predicción entre vistas entre imágenes en la misma unidad de acceso. Al codificar una imagen en una de las vistas no base, la imagen se puede añadir a una lista de imágenes de referencia, si está en una vista diferente pero dentro de la misma instancia de tiempo (por ejemplo, que tiene el mismo valor de POC y, por lo tanto, en la misma unidad de acceso). Una imagen de referencia de predicción entre vistas puede colocarse en cualquier posición de una lista de imágenes de referencia, al igual que cualquier imagen de referencia entre predicciones.

Típicamente, la generación de una lista de imágenes de referencia para la primera o la segunda lista de imágenes de referencia de una imagen B incluye dos etapas: inicialización de lista de imágenes de referencia y reordenación (modificación) de lista de imágenes de referencia. La inicialización de lista de imágenes de referencia es un mecanismo explícito de acuerdo con el cual un codificador de vídeo coloca las imágenes de referencia de la memoria de imágenes de referencia (también conocida como búfer de imágenes descodificadas) en una lista en base al el orden de valores de POC (recuento de orden de imagen, alineado con el orden de visualización una imagen).

El codificador de vídeo puede usar el mecanismo de reordenación de lista de imágenes de referencia para modificar la posición de una imagen que se colocó en la lista durante la inicialización de lista de imágenes de referencia a cualquier nueva posición, o colocar cualquier imagen de referencia de la memoria de imágenes de referencia en cualquier posición, incluso si la imagen no pertenece a la lista inicializada. Algunas imágenes después de la reordenación (modificación) de lista de imágenes de referencia pueden colocarse en una posición adicional en la lista. Sin embargo, si la posición de una imagen excede el número de imágenes de referencia activas de la lista, la imagen no se considera una entrada de la lista de imágenes de referencia final. El número de imágenes de referencia activas puede indicarse en la cabecera de segmento para cada lista. Una vez generadas las listas de imágenes de referencia (por ejemplo, RefPicList0 y RefPicList1, si están disponibles), se puede usar un índice de referencia a una lista de imágenes de referencia para identificar cualquier imagen de referencia incluida en la lista de imágenes de referencia.

Para obtener un predictor de vector de movimiento temporal (TMVP), primero debe identificarse una imagen coubicada. Si la imagen actual es un segmento B, una `collocated_from_I0_flag` se señala en la cabecera de segmento para indicar si la imagen coubicada es de RefPicList0 o RefPicList1. Después de que se identifica una lista de imágenes de referencia, `collocated_ref_idx`, señalado en la cabecera de segmento, se usa para identificar la imagen en la imagen de la lista. A continuación, se identifica una PU coubicada comprobando la imagen coubicada. Se usa el movimiento de la PU inferior derecha de la CU que contiene esta PU, o el movimiento de la PU inferior derecha dentro de las PU centrales de la CU que contiene esta PU. Cuando los vectores de movimiento identificados por el proceso anterior se usan para generar un candidato de movimiento para AMVP o el modo de fusión, es necesario escalarlos en base a la ubicación temporal (reflejada por POC).

En HEVC, el conjunto de parámetros de secuencia (SPS) incluye una bandera `sps_temporal_mvp_enable_flag` y la cabecera de segmento incluye una bandera `pic_temporal_mvp_enable_flag` cuando `sps_temporal_mvp_enable_flag` es igual a 1. Cuando tanto `pic_temporal_mvp_enable_flag` como la identificación temporal son iguales a 0 para una imagen particular, ningún vector de movimiento de las imágenes anteriores a esa imagen particular en el orden de decodificación se usará como predictor de vector de movimiento temporal en la decodificación de la imagen particular o una imagen posterior a la imagen particular en orden de decodificación.

Actualmente, el Grupo de expertos en imágenes en movimiento (MPEG) está desarrollando una norma 3DV basada en HEVC, para lo cual parte de los esfuerzos de normalización también incluye la normalización del códec de vídeo multivista basado en HEVC. De forma similar, en 3DV basado en HEVC, la predicción entre vistas basada en las componentes de vista reconstruida de diferentes vistas está habilitada.

AVC se amplió mediante una ampliación multivista de manera que la ampliación realmente cumpla con el requisito de "solo HLS" (solo sintaxis de alto nivel). El requisito "solo HLS" garantiza que solo haya cambios de sintaxis de alto nivel (HLS) en la codificación de vídeo multivista (MVC), de modo que ningún módulo a nivel de macrobloque en AVC necesita ser rediseñado y puede reutilizarse completamente para MVC. Es posible que se cumpla el requisito de "solo HLS" para una ampliación MVC/3DV de HEVC, y también para la ampliación de codificación de vídeo escalable (SVC) de HEVC, si la descodificación multibucle se considera aceptable.

Para permitir la predicción entre vistas, los cambios de HLS pueden realizarse con el siguiente propósito: identificación de imagen, donde la generación y el marcado de lista de imágenes de referencia deben poder identificar una imagen en una vista específica.

Los cambios HLS no son suficientes para cumplir con el requisito de "solo HLS" en H.264/MVC, ya que se realizan otras restricciones y suposiciones de modo que los módulos de codificación de bajo nivel nunca se encuentren en una situación de, por ejemplo, gestión de movimiento cero relacionado con el escalado. Dichas restricciones, modificaciones y suposiciones son:

- Inhabilitar el modo directo temporal si una imagen coubicada es una imagen de referencia (solo) entre vistas
- Considerar una imagen de referencia (solo) entre vistas como no a corto plazo: en relación con un espacio directo
- Inhabilitar una predicción ponderada implícita

Para cumplir con el requisito de "solo HLS", dichas modificaciones en una ampliación solo deben estar en la sintaxis de alto nivel. Por lo tanto, no debe haber modificaciones para los elementos de sintaxis en la cabecera de segmento, y no debe haber cambios en el proceso de descodificación a nivel de CU para la especificación de ampliación; por ejemplo, la predicción de vectores de movimiento de la especificación de ampliación de HEVC debe ser exactamente la misma que la de la especificación base de HEVC. Los cambios HLS son cambios de descodificador normativos de la especificación de ampliación; sin embargo, desde el punto de vista de la especificación base, dichos cambios no han de ser necesariamente conocidos y pueden ser informativos.

Para habilitar funcionalidades tal como la predicción eficiente entre vistas, se pueden implementar tanto modificaciones en la ampliación de HEVC como en las especificaciones base. Los cambios en la especificación base que no afectan a los procesos de descodificación típicos o a la eficacia de codificación de los descodificadores HEVC base, sino a las funcionalidades habilitadoras de destino en la especificación de ampliación, se denominan ganchos. En la mayoría de los casos, se cumple un requisito de "solo HLS" con ganchos en la especificación base y en los cambios HLS en la especificación de ampliación. Si los ganchos en las especificaciones base no están bien definidos, es posible que determinada funcionalidad deseada no esté habilitada en la especificación de ampliación o que necesiten muchas modificaciones en la especificación de ampliación.

En SVC de solo HLS, una representación de capa base, posiblemente después del muestreo ascendente y/o filtrado, puede introducirse en la lista de imágenes de referencia de la imagen actual de la capa actual. Una imagen de este tipo se denomina imagen de referencia entre capas.

Se pueden realizar diversas modificaciones tanto en la especificación base como en la especificación de ampliación de una modificación HEVC de solo HLS. Dada una determinada funcionalidad deseada, en una fase en la que se pueden modificar los diseños de las especificaciones base y de ampliación, debe haber un equilibrio entre la modificación de la especificación de base y la modificación de la especificación de ampliación.

Las FIGS. 5-9 son diagramas conceptuales que ilustran posibles problemas que deben superarse para lograr una ampliación HEVC de solo HLS. La FIG. 5, por ejemplo, ilustra un ejemplo que no pertenece a la invención en el que una imagen actual 100 incluye bloques, tales como los bloques 102 y 104, predichos usando diversas técnicas de predicción. Específicamente, la imagen actual 100 corresponde a una imagen de una vista no base, mientras que una imagen de referencia entre vistas 110 es una imagen de una vista base. El bloque 102 de la imagen actual 100 se predice entre vistas en relación con la imagen de referencia entre vistas 110 (usando el vector de movimiento de disparidad 106), mientras que el bloque 104 se predice usando interpredicción relativa a la imagen de referencia a corto plazo (ST) 112 de la misma vista no base (usando el vector de movimiento temporal 108). La FIG. 5 ilustra, por tanto, un ejemplo que no pertenece a la invención en el que una imagen actual incluye bloques vecinos con un vector de movimiento temporal (vector de movimiento temporal 108) y un vector de movimiento entre vistas (también denominado vector de movimiento de disparidad, es decir, vector de movimiento de disparidad 106).

Esta divulgación reconoce que, en algunos ejemplos, un vector de movimiento de disparidad no se escalará para predecir un vector de movimiento temporal. Además, esta divulgación también reconoce que, en algunos ejemplos, un vector de movimiento temporal no se escalará para predecir un vector de movimiento de disparidad. Esta

divulgación también reconoce que debería ser posible inhabilitar la predicción de un vector de movimiento de disparidad a partir de un vector de movimiento temporal a corto plazo, por ejemplo, durante AMVP, e inhabilitar la predicción de un vector de movimiento temporal a partir de un vector de movimiento de disparidad. Los vectores de movimiento de disparidad corresponden típicamente a la disparidad local del mismo objeto en diferentes vistas. Sin embargo, los vectores de movimiento temporal corresponden típicamente al movimiento de un objeto. En HTM, que es el software de referencia 3DV, la predicción entre vectores de movimiento de las dos categorías anteriores está inhabilitada.

La FIG. 6 ilustra un ejemplo que no pertenece a la invención en el que una imagen actual incluye bloques predichos usando imágenes de referencia entre vistas de diferentes vistas. Específicamente, en este ejemplo, la imagen de referencia entre vistas 120 está en la vista 0, y la imagen de referencia entre vistas 122 está en la vista 1. La imagen actual 124 está en la vista 2. La imagen actual 124 incluye los bloques 126, 128 predichos, usando predicción entre vistas, tanto de la imagen de referencia entre vistas 120 de la vista 0 como de la imagen de referencia entre vistas 122 de la vista 1. Específicamente, en este ejemplo, el bloque 126 se predice a partir de la imagen de referencia entre vistas 122, mientras que el bloque 128 se predice a partir de la imagen de referencia entre vistas 120.

Los bloques 126 y 128 se predicen usando diferentes vectores de movimiento de disparidad. Es decir, el bloque 126 se predice usando el vector de movimiento de disparidad 130, que hace referencia a una parte de la imagen de referencia entre vistas 122, mientras que el bloque 128 se predice usando el vector de movimiento de disparidad 132, que hace referencia a una parte de la imagen de referencia entre vistas 120. En consecuencia, la FIG. 6 representa un ejemplo en el que una imagen actual incluye bloques vecinos con vectores de movimiento entre vistas que hacen referencia a imágenes de referencia entre vistas de diferentes vistas.

Esta divulgación reconoce que debería ser posible identificar si dos vectores de movimiento de disparidad corresponden a la misma imagen de referencia. Cuando una entrada en RefPicList0 y una entrada en RefPicList1 son imágenes de referencia entre vistas, debería ser posible, durante AMVP, identificar si estas dos imágenes de referencia son iguales. Cuando una RefPicListX (donde 'X' puede representar un valor de 0 o 1, por ejemplo) contiene dos entradas que son imágenes de referencia entre vistas, debería ser posible, durante AMVP, identificar si estas dos imágenes de referencia son iguales. Además, dos entradas con el mismo valor de POC pueden no ser idénticas, por ejemplo, cuando las dos entradas corresponden a vistas diferentes, como se muestra en la FIG. 6.

La FIG. 7 ilustra un ejemplo, que no pertenece a la invención, en el que una imagen actual en una vista no base incluye bloques predichos usando tanto predicción entre vistas relativa a una imagen de referencia entre vistas en una vista base como interpredicción relativa a una imagen de referencia a largo plazo (LT) en la vista no base. Es decir, la FIG. 7 ilustra un ejemplo en el que la imagen actual 140 incluye bloques vecinos 146, 148 con el vector de movimiento temporal 152 (que hace referencia a la imagen de referencia a largo plazo 144) y el vector de movimiento entre vistas 150 (que hace referencia a la imagen de referencia entre vistas 142). El vector de movimiento entre vistas 150 también puede denominarse "vector de movimiento de disparidad 150". Esta divulgación reconoce que debería ser posible inhabilitar la predicción de un vector de movimiento de disparidad, tal como el vector de movimiento de disparidad 150, a partir de un vector de movimiento temporal a largo plazo, tal como el vector de movimiento temporal 152, e inhabilitar la predicción de un vector de movimiento temporal a largo plazo a partir de un vector de movimiento de disparidad.

La FIG. 8 ilustra un ejemplo en el que una imagen actual en una vista no base incluye bloques que se predicen usando interpredicción, tanto a partir de una imagen de referencia a largo plazo (LT) como de una imagen de referencia a corto plazo (ST), de la vista no base. Es decir, la FIG. 8 ilustra un ejemplo en el que la imagen actual 160 incluye bloques vecinos 166, 168 con vectores de movimiento temporal tanto a largo plazo como a corto plazo. Específicamente, el bloque 166 se predice usando el vector de movimiento temporal 170, que hace referencia a la imagen de referencia a largo plazo 162, mientras que el bloque 168 se predice usando el vector de movimiento temporal 172, que hace referencia a la imagen de referencia a corto plazo 164. Por lo tanto, el vector de movimiento temporal 170 puede denominarse vector de movimiento a largo plazo o vector de movimiento temporal a largo plazo, mientras que el vector de movimiento temporal 172 puede denominarse vector de movimiento a corto plazo o vector de movimiento temporal a corto plazo. De acuerdo con la invención, la predicción entre vectores de movimiento temporal a corto plazo y vectores de movimiento temporal a largo plazo está inhabilitada, por ejemplo, durante AMVP.

La FIG. 9 ilustra un ejemplo que no pertenece a la invención en el que una imagen actual en una vista no base incluye bloques que se predicen usando interpredicción, donde los bloques se predicen en relación con diferentes imágenes de referencia a largo plazo (LT) de la vista no base. Es decir, la FIG. 9 ilustra un ejemplo en el que la imagen actual 180 incluye bloques vecinos 186, 188 con vectores de movimiento temporal 190, 192 que hacen referencia a imágenes a largo plazo 184, 182, respectivamente. Específicamente, en este ejemplo, el bloque 186 se predice usando el vector de movimiento temporal 190, que hace referencia a una parte de la imagen de referencia a largo plazo 184, mientras que el bloque 188 se predice usando el vector de movimiento temporal 192, que hace referencia a una parte de la imagen de referencia a largo plazo 182. Esta divulgación reconoce que debería ser posible habilitar y/o inhabilitar la predicción de vectores de movimiento temporal a largo plazo durante AMVP.

La FIG. 10 es un diagrama conceptual que ilustra un conjunto de ejemplo de bloques vecinos a un bloque actual. En particular, en este ejemplo, el bloque actual tiene bloques vecinos a la izquierda etiquetados como A0 y A1 y bloques

vecinos superiores B0, B1 y B2. El bloque actual puede codificarse usando interpredicción, por ejemplo, predicción temporal o predicción entre vistas. Por tanto, un codificador de vídeo, tal como un codificador de vídeo 20 o un decodificador de vídeo 30, puede codificar el bloque actual usando un vector de movimiento. Además, el codificador de vídeo puede codificar el vector de movimiento. En varios ejemplos, el codificador de vídeo puede codificar el vector de movimiento para el bloque actual usando técnicas descritas anteriormente, por ejemplo, para predicción avanzada de vectores de movimiento (AMVP), predicción de vectores de movimiento temporal (TMVP) o modo de fusión. Un predictor TMVP puede corresponder a un vector de movimiento para un bloque que está coubicado con el bloque actual en una imagen codificada previamente.

Los vectores de movimiento de uno o más de los bloques vecinos A0, A1, B0, B1 y B2 pueden ser de diferentes tipos que el vector de movimiento usado para codificar el bloque actual. Por ejemplo, el bloque actual puede codificarse usando un vector de movimiento a largo plazo, mientras que uno o más de los bloques A0, A1, B0, B1 y B2 pueden codificarse usando un vector de movimiento a corto plazo. Como otro ejemplo, el bloque actual puede codificarse usando un vector de movimiento a corto plazo, mientras que uno o más de los bloques A0, A1, B0, B1 y B2 pueden codificarse usando un vector de movimiento a largo plazo. Como otro ejemplo más, el bloque actual puede codificarse usando un vector de movimiento de disparidad, mientras que uno o más de los bloques A0, A1, B0, B1 y B2 pueden codificarse usando un vector de movimiento temporal. Como otro ejemplo más, el bloque actual puede codificarse usando un vector de movimiento temporal, mientras que uno o más de los bloques A0, A1, B0, B1 y B2 pueden codificarse usando un vector de movimiento de disparidad. En tales casos, como se explicó anteriormente, un codificador de vídeo, tal como el codificador de vídeo 20 o el decodificador de vídeo 30, puede inhabilitar la predicción de vectores de movimiento entre vectores de movimiento de diferentes tipos.

El ejemplo de la FIG. 10 ilustra candidatos a predictor de vector de movimiento espacial. Sin embargo, debe entenderse que los candidatos a predictor de vector de movimiento temporal también pueden considerarse para la predicción de vectores de movimiento temporal (TMVP). Dichos candidatos a TMVP pueden corresponder a información de movimiento para bloques coubicados en imágenes previamente codificadas, es decir, bloques que están coubicados con el bloque etiquetado como "bloque actual" en la FIG. 10. Además, de acuerdo con las técnicas de esta divulgación, un candidato a TMVP puede considerarse no disponible para su uso como predictor de vector de movimiento cuando la información de movimiento del candidato a TMVP y el vector de movimiento para el bloque actual apuntan a imágenes de diferentes tipos, por ejemplo, imágenes de referencia a corto y largo plazo.

La FIG. 11 es un diagrama de flujo que ilustra un ejemplo de procedimiento para codificar datos de vídeo de acuerdo con las técnicas de esta divulgación. Los etapas del procedimiento de ejemplo de la FIG. 11 puede, de forma alternativa, realizarse en un orden diferente, o sustancialmente en paralelo, en algunos ejemplos. Asimismo, se pueden omitir determinadas etapas y/o se pueden añadir otras etapas. Aunque se describe que se realiza por el codificador de vídeo 20, debe entenderse que otros dispositivos de codificación de vídeo pueden configurarse para realizar un procedimiento sustancialmente similar.

En este ejemplo, el codificador de vídeo 20 codifica valores de recuento de orden de imagen (POC) de imágenes de referencia para una imagen actual (200). Por ejemplo, el codificador de vídeo 20 codifica valores POC, o datos representativos de los valores POC (tales como bits menos significativos (LSB)) para determinadas imágenes de referencia en una estructura de datos de conjunto de parámetros de secuencia (SPS) para una secuencia que incluye la imagen actual. El codificador de vídeo 20 también puede codificar, de forma adicional o alternativa, valores de POC para una o más imágenes de referencia en una cabecera de segmento de un segmento actual de la imagen actual. En algunos ejemplos, el codificador de vídeo 20 puede codificar datos que representan valores de POC de imágenes de referencia a largo plazo en un SPS y valores de POC de imágenes de referencia a corto plazo en una cabecera de segmento. El codificador de vídeo 20 también puede codificar valores de POC de imágenes de referencia entre vistas, por ejemplo, en el SPS, la cabecera de segmento o en cualquier otro lugar. En general, los valores de POC de las imágenes de referencia entre vistas son los mismos que el valor de POC de la imagen actual que se está codificando.

El codificador de vídeo 20 también codifica identificadores de segunda dimensión de las imágenes de referencia (202). Los identificadores de segunda dimensión pueden incluir uno o más identificadores de vista para vistas que incluyen las imágenes de referencia, índices de orden de vista para las vistas que incluyen las imágenes de referencia, una combinación de los índices de orden de vista y banderas de profundidad, identificadores de capa para capas de codificación de vídeo escalable (SVC) que incluyen las imágenes de referencia y/o identificadores de capa genéricos. De esta manera, la combinación de un valor de POC para una imagen de referencia y el identificador de segunda dimensión para la imagen de referencia se usa para identificar la imagen de referencia.

El codificador de vídeo 20 realiza una búsqueda de movimiento para un bloque actual de la imagen actual. Es decir, la unidad de estimación de movimiento 42 busca en las imágenes de referencia el bloque de referencia que se asemeje más al bloque actual. Esto da como resultado información de movimiento, incluido un vector de movimiento, que hace referencia al bloque de referencia, así como a la imagen de referencia en la que aparece el bloque de referencia. Por tanto, la unidad de compensación de movimiento 44 del codificador de vídeo 20 puede predecir el bloque actual usando el vector de movimiento que apunta a una de las imágenes de referencia (204).



El codificador de vídeo 20 también puede codificar el vector de movimiento, por ejemplo, usando predicción avanzada de vector de movimiento (AMVP), predicción de vectores de movimiento temporal (TMVP) o el modo de fusión. En particular, el codificador de vídeo 20 puede determinar un conjunto de predictores de vector de movimiento candidatos disponibles (206). Por ejemplo, haciendo referencia a la FIG. 10, el codificador de vídeo 20 puede determinar si los vectores de movimiento para los bloques vecinos A0, A1, B0, B1 y B2 están disponibles. En particular, de acuerdo con las técnicas de esta divulgación, el codificador de vídeo 20 puede determinar que un vector de movimiento de uno de estos bloques vecinos no está disponible cuando el vector de movimiento del bloque vecino es de un tipo diferente al vector de movimiento para el bloque actual. De manera similar, el codificador de vídeo 20 puede determinar si un vector de movimiento para un candidato a predictor de vector de movimiento temporal hace referencia a un tipo diferente de imagen de referencia que el vector de movimiento para el bloque actual para determinar si el candidato a TMVP está disponible para su uso como predictor para codificar el vector de movimiento del bloque actual.

Como se explicó anteriormente, los ejemplos de diferentes tipos de vectores de movimiento incluyen vectores de movimiento a largo plazo, vectores de movimiento a corto plazo, vectores de movimiento temporal y vectores de movimiento de disparidad. Por lo tanto, el codificador de vídeo 20 puede determinar un tipo para el vector de movimiento del bloque actual, así como tipos para los vectores de movimiento de los bloques vecinos, y determinar que los vectores de movimiento de los bloques vecinos de tipos diferentes al tipo para el vector de movimiento actual del bloque actual no están disponibles para su uso como predictores de vectores de movimiento para el vector de movimiento actual. Para determinar los tipos, el codificador de vídeo 20 puede hacer referencia a valores de POC de las imágenes de referencia a las que hacen referencia los vectores de movimiento candidatos, el valor de POC de la imagen de referencia a la que hace referencia el vector de movimiento actual, los identificadores de segunda dimensión de las imágenes de referencia a las que hacen referencia los vectores de movimiento candidatos, y/o el identificador de segunda dimensión de la imagen de referencia a la que hace referencia el vector de movimiento actual.

Posteriormente, el codificador de vídeo 20 puede seleccionar uno de los predictores de vector de movimiento candidatos disponibles de un bloque vecino (que puede incluir un bloque cópico en una imagen codificada previamente y/o un bloque correspondiente en una imagen de una vista diferente) como predictor de vector de movimiento para el vector de movimiento actual (208). El codificador de vídeo 20 puede codificar entonces el vector de movimiento actual usando el predictor de vector de movimiento seleccionado (210).

Además, el codificador de vídeo 20 puede calcular un bloque residual para el bloque actual (212). Como se explica con respecto a la FIG. 2, el sumador 50 puede calcular diferencias píxel a píxel entre el bloque original no codificado y el bloque predicho formado por la unidad de compensación de movimiento 44. La unidad de procesamiento de transformada 52, la unidad de cuantificación 54 y la unidad de codificación entrópica 56 pueden entonces, respectivamente, transformar, cuantificar y explorar el bloque residual (214). Específicamente, la unidad de procesamiento de transformada 52 puede transformar el bloque residual para producir un bloque de coeficientes de transformada, la unidad de cuantificación 52 puede cuantificar los coeficientes de transformada y la unidad de codificación entrópica 56 puede explorar los coeficientes de transformada cuantificados. La unidad de codificación entrópica 56 puede entonces codificar entrópicamente los coeficientes de transformada cuantificados y la información de vector de movimiento codificada (216).

De esta manera, el procedimiento de la FIG. 11 representa un ejemplo de un procedimiento que incluye codificar un valor de recuento de orden de imagen (POC) para una primera imagen de datos de vídeo, codificar un identificador de imagen de segunda dimensión para la primera imagen y codificar, de acuerdo con una especificación base de codificación de vídeo (o una ampliación de la especificación base de codificación de vídeo), una segunda imagen en base a, al menos en parte, el valor de POC y el identificador de imagen de segunda dimensión de la primera imagen. Además, el procedimiento puede incluir inhabilitar la predicción de vectores de movimiento entre un primer vector de movimiento de un primer bloque de la segunda imagen, en el que el primer vector de movimiento hace referencia a una imagen de referencia a corto plazo, y un segundo vector de movimiento de un segundo bloque de la segunda imagen, en el que el segundo vector de movimiento hace referencia a una imagen de referencia a largo plazo. De forma adicional o alternativa, el procedimiento puede incluir inhabilitar la predicción de vectores de movimiento entre un primer vector de movimiento de un primer bloque de la segunda imagen, en el que el primer vector de movimiento hace referencia a una imagen de referencia entre vistas, y un segundo vector de movimiento de un segundo bloque de la segunda imagen, en el que el segundo vector de movimiento hace referencia a una imagen de referencia temporal.

La FIG. 12 es un diagrama de flujo que ilustra un ejemplo de procedimiento para descodificar datos de vídeo de acuerdo con las técnicas de esta divulgación. Los etapas del procedimiento de ejemplo de la FIG. 12 puede, de forma alternativa, realizarse en un orden diferente, o sustancialmente en paralelo, en algunos ejemplos. Asimismo, se pueden omitir determinadas etapas y/o se pueden añadir otras etapas. Aunque se describe que se realiza por el descodificador de vídeo 30, debe entenderse que otros dispositivos de descodificación de vídeo pueden configurarse para realizar un procedimiento sustancialmente similar.

En este ejemplo, el descodificador de vídeo 30 descodifica los valores de POC de imágenes de referencia para una imagen actual (230). Por ejemplo, el descodificador de vídeo 30 puede descodificar valores POC, o datos representativos de los valores POC (tales como bits menos significativos (LSB)) para determinadas imágenes de

referencia en una estructura de datos de conjunto de parámetros de secuencia (SPS) para una secuencia que incluye la imagen actual. El decodificador de vídeo 30 puede reconstruir los valores de POC a partir de LSB descodificados para los valores de POC añadiendo los LSB a los respectivos MSB obtenidos de, por ejemplo, un valor de POC completo previamente descodificado. El decodificador de vídeo 30 también puede descodificar, de forma adicional o alternativa, valores de POC para una o más imágenes de referencia en una cabecera de segmento de un segmento actual de la imagen actual. En algunos ejemplos, el decodificador de vídeo 30 puede descodificar datos que representan valores de POC de imágenes de referencia a largo plazo en un SPS y valores de POC de imágenes de referencia a corto plazo en una cabecera de segmento. El decodificador de vídeo 30 también puede descodificar valores de POC de imágenes de referencia entre vistas, por ejemplo, en el SPS, la cabecera de segmento o en cualquier otro lugar. En general, los valores de POC de las imágenes de referencia entre vistas son los mismos que el valor de POC de la imagen actual que se está codificando.

El decodificador de vídeo 30 también puede descodificar identificadores de segunda dimensión de las imágenes de referencia (232). Los identificadores de segunda dimensión pueden incluir uno o más identificadores de vista para vistas que incluyen las imágenes de referencia, índices de orden de vista para las vistas que incluyen las imágenes de referencia, una combinación de los índices de orden de vista y banderas de profundidad, identificadores de capa para capas de codificación de vídeo escalable (SVC) que incluyen las imágenes de referencia y/o identificadores de capa genéricos. De esta manera, la combinación de un valor de POC para una imagen de referencia y el identificador de segunda dimensión para la imagen de referencia se puede usar para identificar la imagen de referencia. Por tanto, para identificar una imagen de referencia, la información de movimiento puede incluir tanto un valor de POC como un identificador de segunda dimensión para la imagen de referencia.

El decodificador de vídeo 30 también puede descodificar un vector de movimiento para un bloque actual de la imagen actual. En particular, el decodificador de vídeo 30 puede determinar un conjunto de predictores de vector de movimiento candidatos disponibles (234). Por ejemplo, haciendo referencia a la FIG. 10, el decodificador de vídeo 30 puede determinar si los vectores de movimiento para los bloques vecinos A0, A1, B0, B1 y B2 están disponibles. En particular, de acuerdo con las técnicas de esta divulgación, el decodificador de vídeo 30 puede determinar que un vector de movimiento de uno de estos bloques vecinos no está disponible cuando el vector de movimiento del bloque vecino es de un tipo diferente al vector de movimiento para el bloque actual. De manera similar, el decodificador de vídeo 30 puede determinar si un vector de movimiento para un candidato a predictor de vector de movimiento temporal hace referencia a un tipo diferente de imagen de referencia que el vector de movimiento para el bloque actual para determinar si el candidato a TMVP está disponible para su uso como predictor para codificar el vector de movimiento del bloque actual.

Como se explicó anteriormente, los ejemplos de diferentes tipos de vectores de movimiento incluyen vectores de movimiento a largo plazo, vectores de movimiento a corto plazo, vectores de movimiento temporal y vectores de movimiento de disparidad. Por lo tanto, el decodificador de vídeo 30 puede determinar un tipo para el vector de movimiento del bloque actual, así como tipos para los vectores de movimiento de los bloques vecinos, y determinar que los vectores de movimiento de los bloques vecinos de tipos diferentes al tipo para el vector de movimiento actual del bloque actual no están disponibles para su uso como predictores de vectores de movimiento para el vector de movimiento actual. Para determinar los tipos, el decodificador de vídeo 30 puede hacer referencia a valores de POC de las imágenes de referencia a las que hacen referencia los vectores de movimiento candidatos, el valor de POC de la imagen de referencia a la que hace referencia el vector de movimiento actual, los identificadores de segunda dimensión de las imágenes de referencia a las que hacen referencia los vectores de movimiento candidatos, y/o el identificador de segunda dimensión de la imagen de referencia a la que hace referencia el vector de movimiento actual.

Posteriormente, el decodificador de vídeo 30 puede seleccionar uno de los predictores de vector de movimiento candidatos disponibles de un bloque vecino (que puede incluir un bloque cúbico en una imagen codificada previamente y/o un bloque correspondiente en una imagen de una vista diferente) como predictor de vector de movimiento para el vector de movimiento actual (236). El decodificador de vídeo 30 puede descodificar entonces el vector de movimiento actual usando el predictor de vector de movimiento seleccionado (238). Por ejemplo, usando AMVP, el decodificador de vídeo 30 puede descodificar valores de diferencia de vector de movimiento (MVD) para el vector de movimiento actual y, a continuación, aplicar los valores de MVD al predictor de vector de movimiento seleccionado. Es decir, el decodificador de vídeo 30 puede añadir una componente x del valor de MVD a una componente x del predictor de vector de movimiento seleccionado, y una componente y del valor de MVD a una componente y del predictor de vector de movimiento seleccionado.

La unidad de compensación de movimiento 72 del decodificador de vídeo 30 puede predecir entonces el bloque actual usando el vector de movimiento, que apunta a una de las imágenes de referencia (240). Es decir, además del propio vector de movimiento, el decodificador de vídeo 30 puede descodificar información de identificación de imagen de referencia para el bloque al que corresponde el vector de movimiento, tal como un valor de POC y un valor de identificación de segunda dimensión. De esta manera, el decodificador de vídeo 30 puede determinar la imagen de referencia a la que apunta el vector de movimiento usando el valor de POC y el valor de identificación de segunda dimensión. Por consiguiente, la unidad de compensación de movimiento 72 puede formar un bloque predicho para el bloque actual usando el vector de movimiento y la información de identificación de imagen de referencia, es decir, el valor de POC y el valor de identificación de segunda dimensión.

La unidad de descodificación entrópica 70 puede además descodificar entrópicamente coeficientes de transformada cuantificados para un bloque residual correspondiente al bloque actual (242). La unidad de descodificación entrópica 70, la unidad de cuantificación inversa 76 y la unidad de transformada inversa 78, respectivamente, exploran, cuantifican y transforman de manera inversa los coeficientes de transformada cuantificados para reproducir el bloque residual (244). El sumador 80 del descodificador de vídeo 30 puede entonces combinar (es decir, añadir, píxel por píxel) el bloque predicho y el bloque residual, para reproducir el bloque actual (246).

De esta manera, el procedimiento de la FIG. 12 representa un procedimiento que incluye descodificar un valor de recuento de orden de imagen (POC) para una primera imagen de datos de vídeo, descodificar un identificador de imagen de segunda dimensión para la primera imagen y descodificar, de acuerdo con una especificación base de codificación de vídeo (o una ampliación de la especificación base de codificación de vídeo), una segunda imagen en base a, al menos en parte, el valor de POC y el identificador de imagen de segunda dimensión de la primera imagen. El procedimiento incluye inhabilitar la predicción de vectores de movimiento entre un primer vector de movimiento de un primer bloque de la segunda imagen, en el que el primer vector de movimiento hace referencia a una imagen de referencia a corto plazo, y un segundo vector de movimiento de un segundo bloque de la segunda imagen, en el que el segundo vector de movimiento hace referencia a una imagen de referencia a largo plazo. De una forma adicional o alternativa que no pertenece a la invención, el procedimiento puede incluir inhabilitar la predicción de vectores de movimiento entre un primer vector de movimiento de un primer bloque de la segunda imagen, en el que el primer vector de movimiento hace referencia a una imagen de referencia entre vistas, y un segundo vector de movimiento de un segundo bloque de la segunda imagen, en el que el segundo vector de movimiento hace referencia a una imagen de referencia temporal.

Se ha de reconocer que dependiendo del ejemplo, determinadas acciones o eventos de cualquiera de las técnicas descritas en el presente documento se pueden realizar en una secuencia diferente, se pueden añadir, fusionar u omitir por completo (por ejemplo, no todas las acciones o eventos descritos son necesarios para la puesta en práctica de las técnicas). Además, en determinados ejemplos, las acciones o los eventos se pueden realizar de forma concurrente, por ejemplo, a través de procesamiento multihilo, 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 cualquier combinación de los mismos. Si se implementan en software, las funciones se pueden almacenar en, o transmitir a través de, un medio legible por ordenador como una o más instrucciones o código, y ejecutar mediante una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible tal como medios de almacenamiento de datos, o medios de comunicación que incluyen cualquier medio que 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) medios de almacenamiento tangibles legibles por ordenador que no son transitorios o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que se puede acceder mediante uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en esta 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 código de programa deseado en forma de instrucciones o estructuras de datos y al que se puede acceder mediante un ordenador. Además, cualquier conexión se denomina apropiadamente medio legible por ordenador. Por ejemplo, si se transmiten instrucciones desde un sitio web, un servidor o cualquier 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 tecnologías inalámbricas tales como infrarrojos, radio y microondas están incluidos en la definición de medio. Sin embargo, se debe entender que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que, en cambio, se refieren a medios de almacenamiento tangibles no transitorios. Los discos, como se usan en el presente documento, incluyen discos compactos (CD), discos láser, discos ópticos, discos versátiles digitales (DVD), disquetes y discos Blu-ray, donde algunos discos normalmente reproducen datos magnéticamente, mientras que otros reproducen datos ópticamente con láseres. Las combinaciones de lo anterior también se deben incluir dentro del alcance de medios legibles por ordenador.

Las instrucciones pueden ser ejecutadas por uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), formaciones de lógica programable *in situ* (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 software dedicados, configurados para la codificación y la descodificación, o incorporarse en un códec combinado. Asimismo, las técnicas se podrían implementar por completo en uno o más circuitos o elementos lógicos.

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

## REIVINDICACIONES

1. Un procedimiento para decodificar datos de vídeo tridimensionales y/o multicapa usando una ampliación solamente de sintaxis de alto nivel de una especificación base de codificación de vídeo bidimensional y de una sola capa, comprendiendo el procedimiento:

descodificar (230), en una estructura de datos de conjunto de parámetros de secuencia, SPS, para una secuencia que incluye una imagen actual, valores de recuento de orden de imagen, POC, para imágenes de referencia de la secuencia y, opcionalmente, decodificar, en una cabecera de segmento de un segmento actual de la imagen actual, valores de POC para imágenes de referencia de la imagen actual;

descodificar (232) un identificador de imagen de segunda dimensión para cada una de las imágenes de referencia, en el que un identificador de imagen de segunda dimensión y un valor de POC identifican conjuntamente una imagen de referencia de los datos de vídeo tridimensionales y/o multicapa; y

descodificar, de acuerdo con la especificación base de codificación de vídeo, la imagen actual, que comprende:

determinar un primer vector de movimiento de un primer bloque de la imagen actual e identificar una primera imagen de referencia a la que hace referencia el primer vector de movimiento en base a, al menos en parte, un primer valor de POC decodificado y un primer identificador de imagen de segunda dimensión decodificado;

determinar un segundo vector de movimiento de un segundo bloque de la imagen actual e identificar una segunda imagen de referencia a la que hace referencia el segundo vector de movimiento en base a, al menos en parte, un segundo valor de POC decodificado y un segundo identificador de imagen de segunda dimensión decodificado;

determinar el tipo del primer y segundo vectores de movimiento usando dichos primer y segundo valores de POC decodificados e identificadores de imagen de segunda dimensión, en el que diferentes tipos de vectores de movimiento incluyen vectores de movimiento temporal a largo plazo, vectores de movimiento temporal a corto plazo y vectores de movimiento de disparidad; e

inhabilitar la predicción de vectores de movimiento entre el primer vector de movimiento y el segundo vector de movimiento cuando el primer vector de movimiento hace referencia a una imagen de referencia a corto plazo y el segundo vector de movimiento hace referencia a una imagen de referencia a largo plazo.

2. Un procedimiento para codificar datos de vídeo tridimensionales y/o multicapa usando una ampliación solamente de sintaxis de alto nivel de una especificación base de codificación de vídeo bidimensional y de una sola capa, comprendiendo el procedimiento:

codificar (200), en una estructura de datos de conjunto de parámetros de secuencia, SPS, para una secuencia que incluye una imagen actual, un valor de recuento de orden de imagen, POC, para imágenes de referencia de los datos de vídeo y, opcionalmente, codificar, en una cabecera de segmento de un segmento actual de la imagen actual, valores de POC para imágenes de referencia de la imagen actual;

codificar (202) un identificador de imagen de segunda dimensión para las imágenes de referencia, en el que el identificador de imagen de segunda dimensión y el valor de POC identifican conjuntamente una imagen de referencia de los datos de vídeo tridimensionales y/o multicapa; y

codificar, de acuerdo con la especificación base de codificación de vídeo, una imagen actual, lo que comprende:

determinar un primer vector de movimiento de un primer bloque de la imagen actual e identificar una primera imagen de referencia a la que hace referencia el primer vector de movimiento en base a, al menos en parte, el valor de POC y el identificador de imagen de segunda dimensión de la primera imagen de referencia; determinar un segundo vector de movimiento de un segundo bloque de la imagen actual e identificar una segunda imagen de referencia a la que hace referencia el segundo vector de movimiento en base a, al menos en parte, el valor de POC y el identificador de imagen de segunda dimensión de la segunda imagen de referencia;

determinar el tipo del primer y segundo vectores de movimiento usando los valores de POC y los identificadores de imagen de segunda dimensión de la primera y segunda imágenes de referencia, en el que diferentes tipos de vectores de movimiento incluyen vectores de movimiento temporal a largo plazo, vectores de movimiento temporal a corto plazo, y vectores de movimiento de disparidad; e

inhabilitar la predicción de vectores de movimiento entre el primer vector de movimiento y el segundo vector de movimiento cuando el primer vector de movimiento hace referencia a una imagen de referencia a corto plazo y el segundo vector de movimiento hace referencia a una imagen de referencia a largo plazo.

3. El procedimiento de la reivindicación 1 o 2, en el que identificar la primera imagen comprende identificar la imagen de referencia durante, respectivamente, la decodificación o codificación del vector de movimiento para un bloque de la imagen actual, en el que la decodificación o codificación del vector de movimiento comprende la decodificación o codificación del vector de movimiento de acuerdo con al menos uno de entre predicción avanzada de vectores de movimiento, AMVP, predicción de vectores de movimiento temporal, TMVP, y modo de fusión de HEVC.

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

habilitar la predicción de vectores de movimiento entre el primer vector de movimiento y el segundo vector de movimiento cuando el primer vector de movimiento hace referencia a una imagen de referencia a corto plazo y el segundo vector de movimiento hace referencia a una imagen de referencia a corto plazo; y  
 5 escalar al menos uno del primer vector de movimiento y el segundo vector de movimiento en base a un valor de POC para la primera imagen de referencia a la que hace referencia el primer vector de movimiento y un valor de POC para la segunda imagen de referencia a la que hace referencia el segundo vector de movimiento.

5. El procedimiento de la reivindicación 1 o 2, en el que la especificación base de codificación de vídeo comprende una especificación base de codificación de vídeo de alta eficacia, HEVC, y en el que la ampliación para la  
 10 especificación base de codificación de vídeo comprende una de entre una ampliación de codificación de vídeo escalable, SVC, para la especificación base de HEVC y una ampliación de codificación de vídeo multivista, MVC, para la especificación base de HEVC.

6. El procedimiento de la reivindicación 1 o 2, en el que el identificador de imagen de segunda dimensión comprende al menos uno de entre un identificador de vista para una vista que incluye la primera imagen, un índice de orden de  
 15 vista para la vista que incluye la primera imagen, una combinación del índice de orden de vista y una bandera de profundidad, un identificador de capa para una capa de codificación de vídeo escalable, SVC, que incluye la primera imagen, y un identificador de capa genérico.

7. Un dispositivo (30) para descodificar datos de vídeo tridimensionales y/o multicapa usando una ampliación solamente de sintaxis de alto nivel de una especificación base de codificación de vídeo bidimensional y de una sola  
 20 capa, comprendiendo el dispositivo medios descodificadores de vídeo (70) configurados para:

descodificar, en una estructura de datos de conjunto de parámetros de secuencia, SPS, para una secuencia que  
 25 incluye una imagen actual, valores de recuento de orden de imagen, POC, para imágenes de referencia de la secuencia y, opcionalmente, descodificar, en una cabecera de segmento de un segmento actual de la imagen actual, valores de POC para imágenes de referencia de la imagen actual;  
 descodificar un identificador de imagen de segunda dimensión para cada una de las imágenes de referencia, en el  
 30 que un identificador de imagen de segunda dimensión y un valor de POC identifican conjuntamente una imagen de referencia de los datos de vídeo tridimensionales y/o multicapa; y  
 descodificar la imagen actual de acuerdo con la especificación base de codificación de vídeo, en el que la descodificación de la imagen actual comprende:

determinar un primer vector de movimiento de un primer bloque de la imagen actual e identificar una primera imagen  
 35 de referencia a la que hace referencia el primer vector de movimiento en base a, al menos en parte, un primer valor de POC descodificado y un primer identificador de imagen de segunda dimensión descodificado; determinar un segundo vector de movimiento de un segundo bloque de la imagen actual e identificar una segunda imagen de referencia a la que hace referencia el segundo vector de movimiento en base a, al menos en parte, un segundo valor de POC descodificado y un segundo identificador de imagen de segunda dimensión descodificado;  
 40 determinar el tipo del primer y segundo vectores de movimiento usando dichos primer y segundo valores de POC descodificados e identificadores de imagen de segunda dimensión, en el que diferentes tipos de vectores de movimiento incluyen vectores de movimiento temporal a largo plazo, vectores de movimiento temporal a corto plazo y vectores de movimiento de disparidad; e  
 45 inhabilitar la predicción de vectores de movimiento entre el primer vector de movimiento y el segundo vector de movimiento cuando el primer vector de movimiento hace referencia a una imagen de referencia a corto plazo y el segundo vector de movimiento hace referencia a una imagen de referencia a largo plazo.

8. Un dispositivo (20) para codificar datos de vídeo tridimensionales y/o multicapa usando una ampliación solamente de sintaxis de alto nivel de una especificación base de codificación de vídeo bidimensional y de una sola capa,  
 50 comprendiendo el dispositivo:

medios (50) para codificar, en una estructura de datos de conjunto de parámetros de secuencia, SPS, para una secuencia que incluye una imagen actual, un valor de recuento de orden de imagen, POC, para imágenes de referencia de los datos de vídeo y, opcionalmente, codificar, en una cabecera de segmento de un segmento actual de la imagen  
 55 actual, valores de POC para imágenes de referencia de la imagen actual;  
 medios (56) para codificar un identificador de imagen de segunda dimensión para las imágenes de referencia, en los que el identificador de imagen de segunda dimensión y un valor de POC identifican conjuntamente una imagen de referencia de los datos de vídeo tridimensionales y/o multicapa; y  
 medios para codificar, de acuerdo con la especificación base de codificación de vídeo, una imagen actual, que  
 60 comprenden medios para:

determinar un primer vector de movimiento de un primer bloque de la imagen actual e identificar una primera imagen de referencia a la que hace referencia el primer vector de movimiento en base a, al menos en parte, el valor de POC y el identificador de imagen de segunda dimensión de la primera imagen;

determinar un segundo vector de movimiento de un segundo bloque de la imagen actual e identificar una segunda imagen de referencia a la que hace referencia el segundo vector de movimiento en base a, al menos en parte, el valor de POC y el identificador de imagen de segunda dimensión de la segunda imagen de referencia;

5 determinar el tipo del primer y segundo vectores de movimiento usando los valores de POC y los identificadores de imagen de segunda dimensión de la primera y segunda imágenes de referencia, en el que diferentes tipos de vectores de movimiento incluyen vectores de movimiento temporal a largo plazo, vectores de movimiento temporal a corto plazo, y vectores de movimiento de disparidad; e

10 inhabilitar la predicción de vectores de movimiento entre el primer vector de movimiento y el segundo vector de movimiento cuando el primer vector de movimiento hace referencia a una imagen de referencia a corto plazo y el segundo vector de movimiento hace referencia a una imagen de referencia a largo plazo.

9. Un medio de almacenamiento legible por ordenador que tiene almacenadas en el mismo instrucciones que, cuando se ejecutan, hacen que un procesador lleve a cabo el procedimiento de cualquiera de las reivindicaciones 1 a 6.

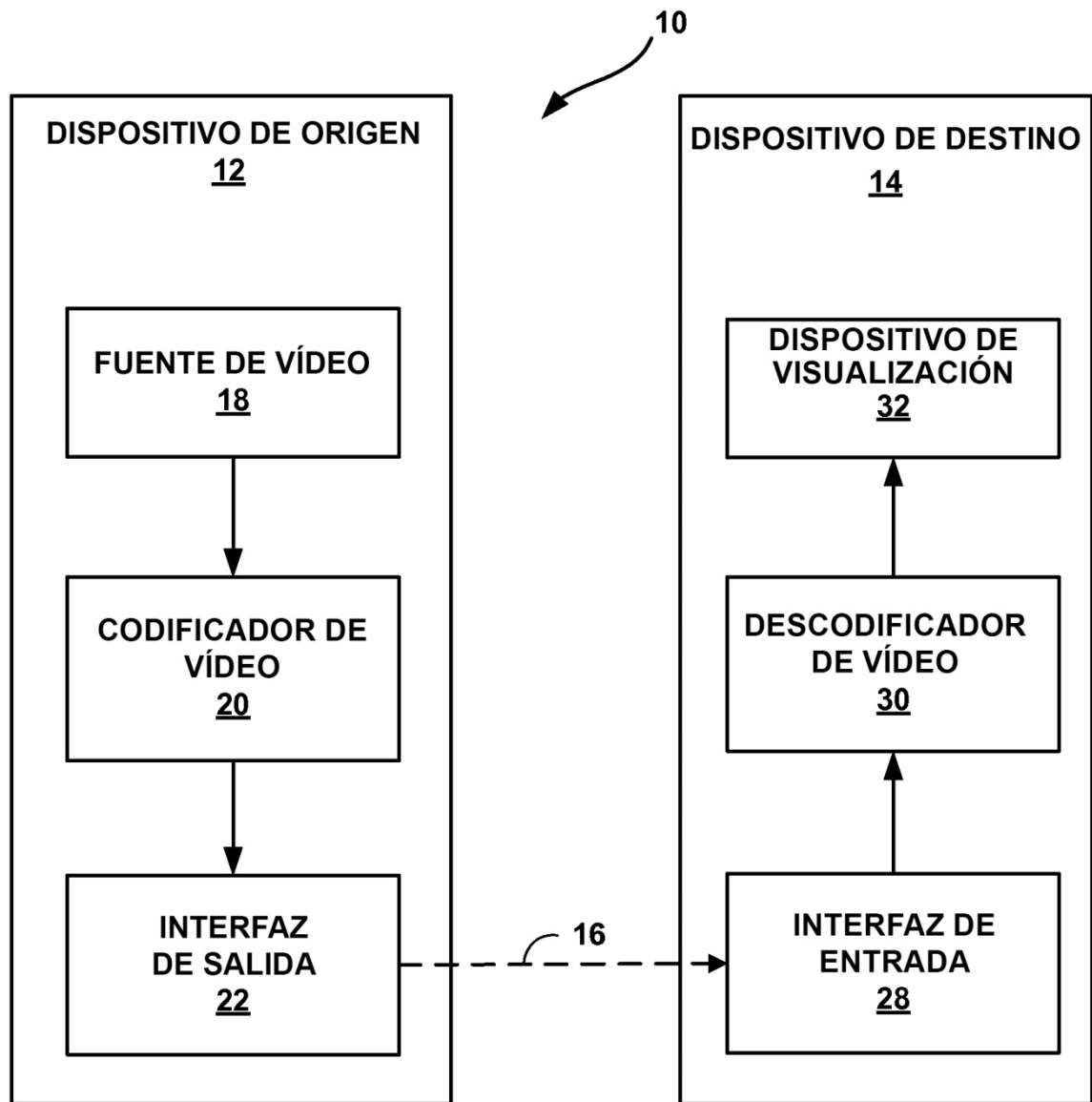


FIG. 1



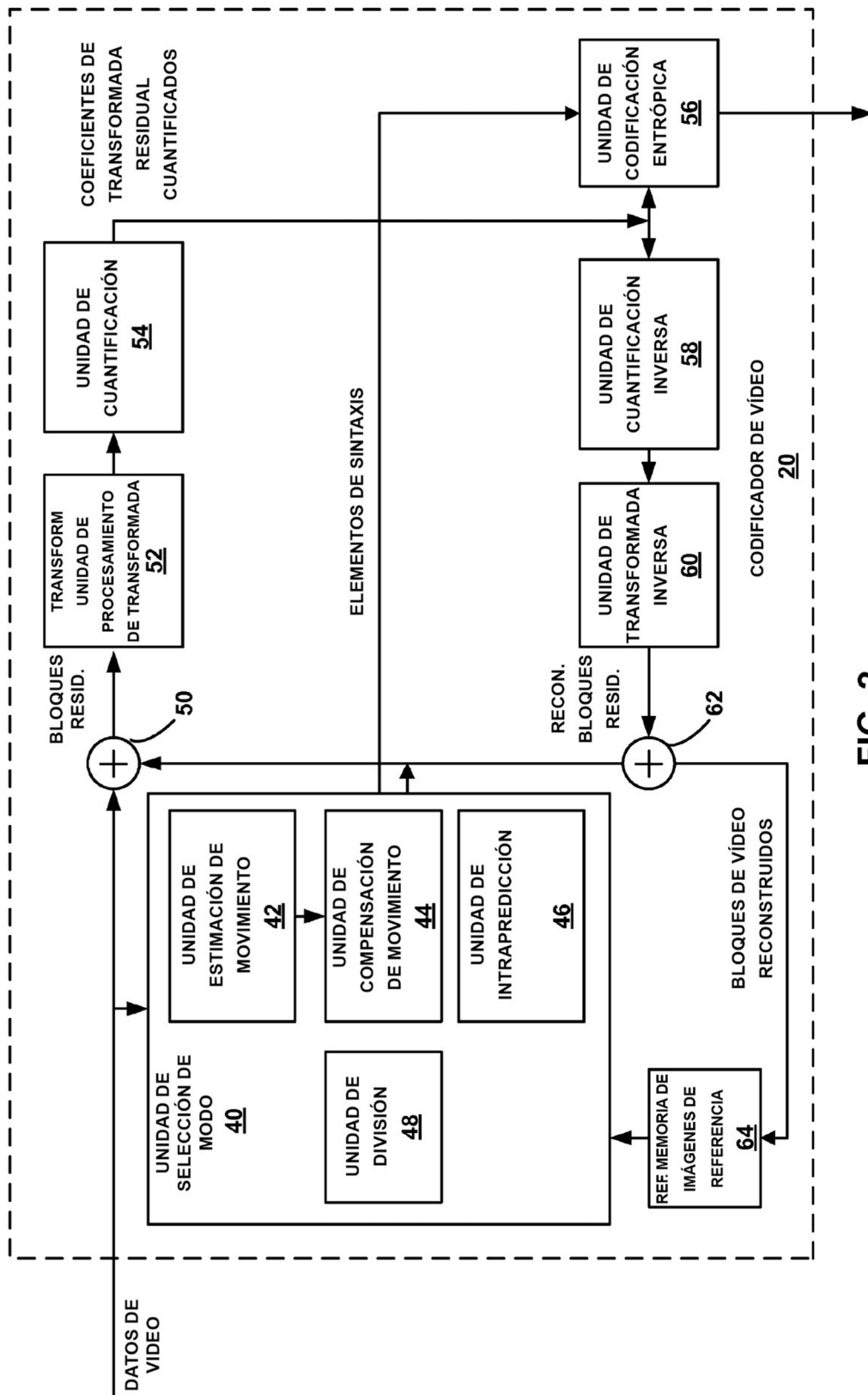


FIG. 2

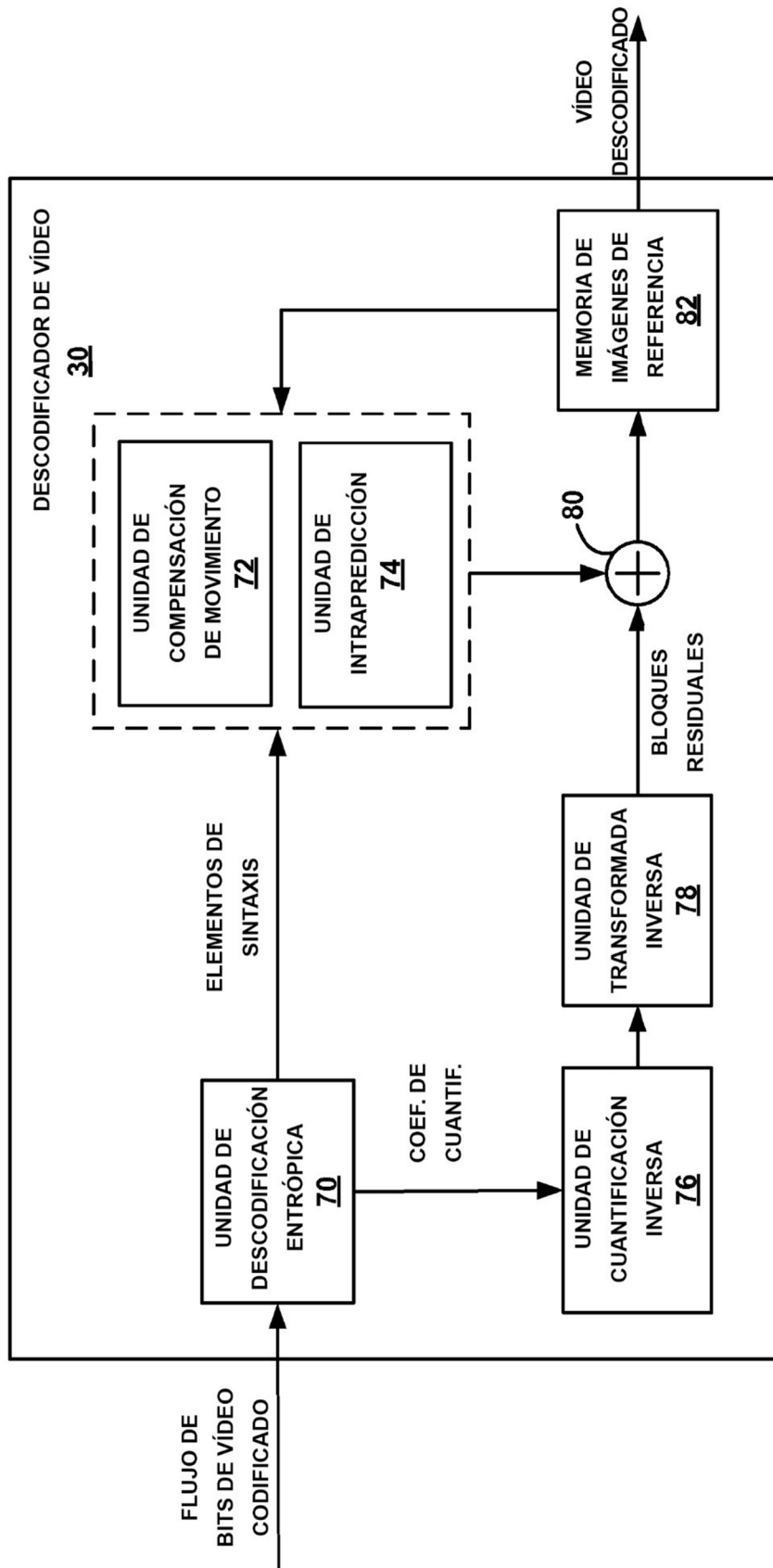


FIG. 3

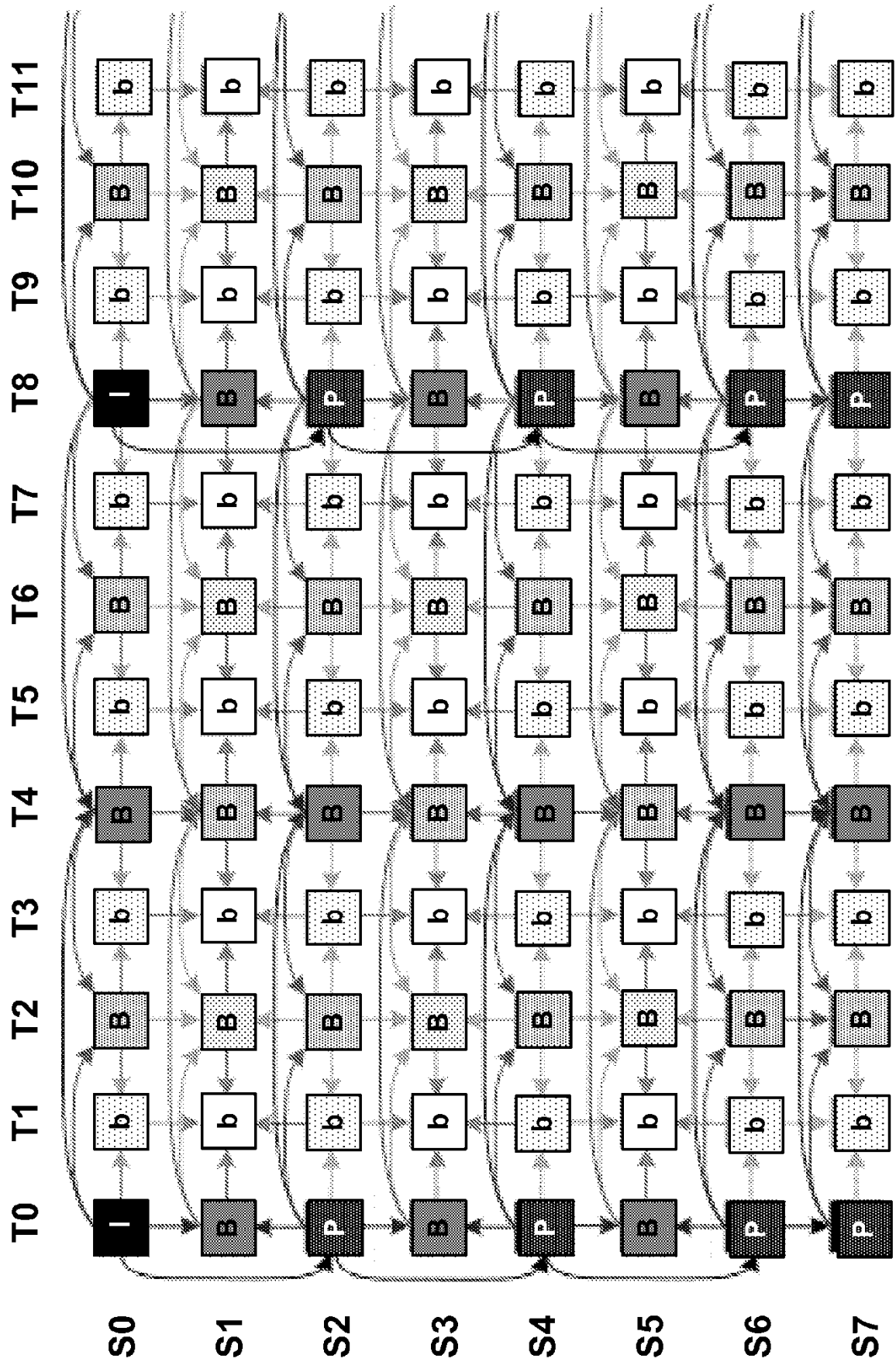


FIG. 4

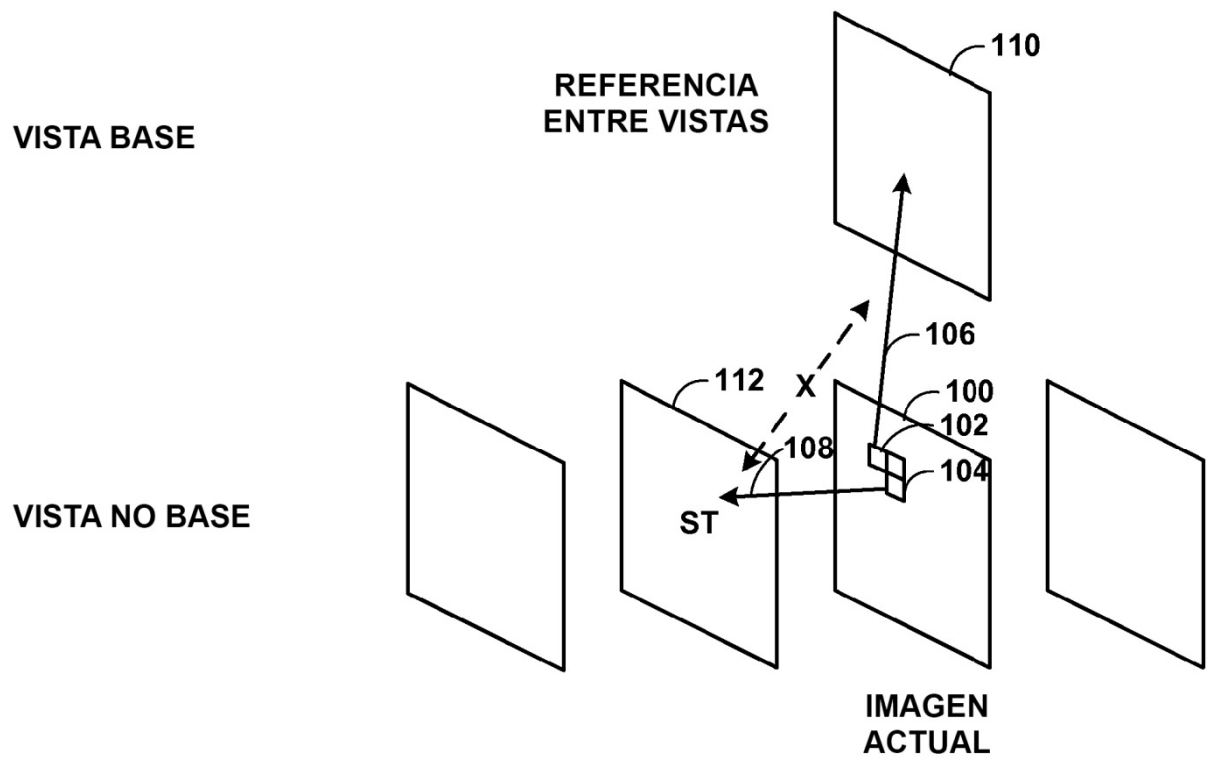
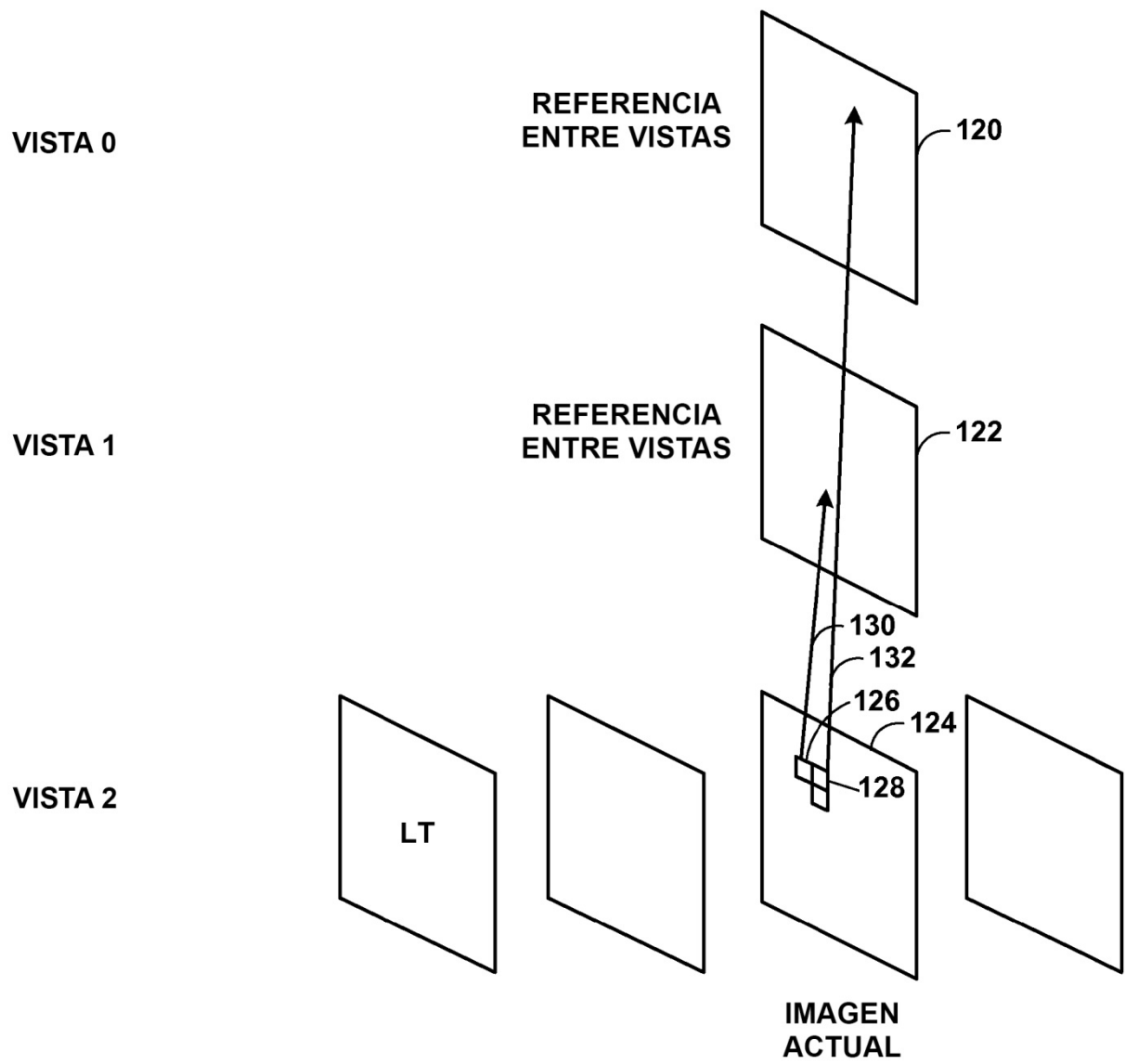
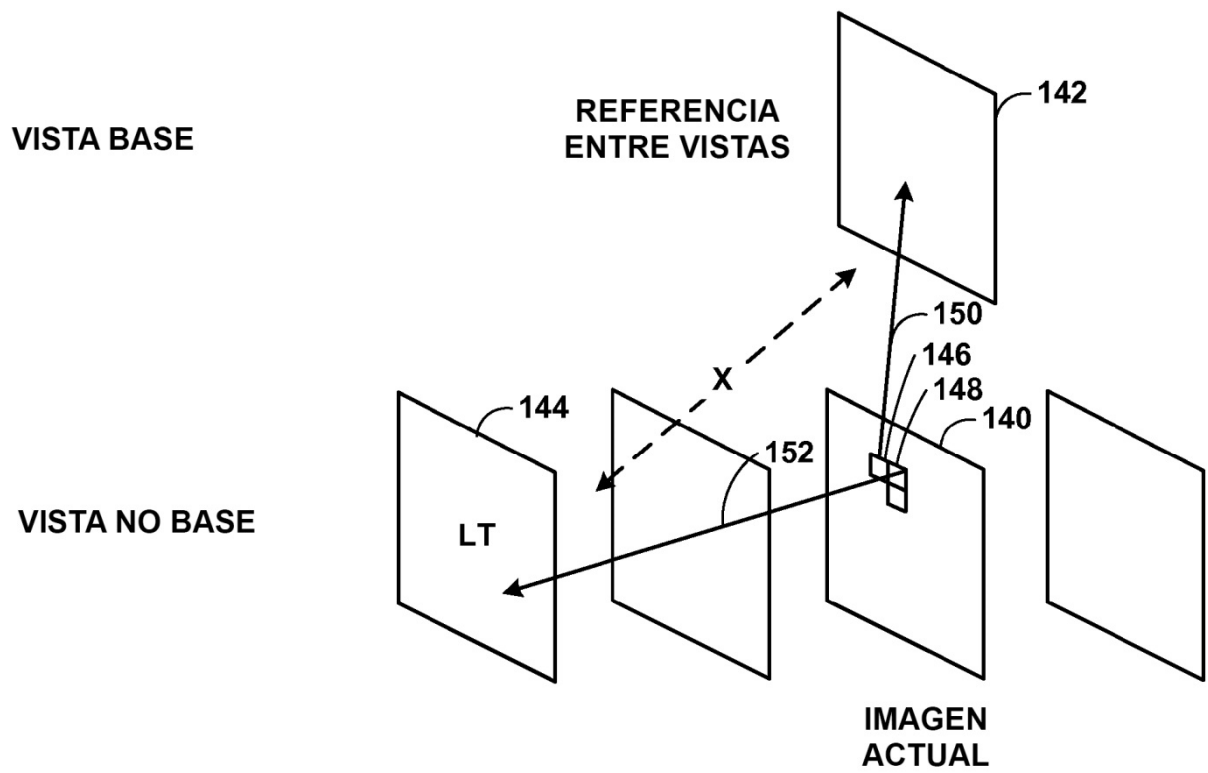


FIG. 5



**FIG. 6**



**FIG. 7**

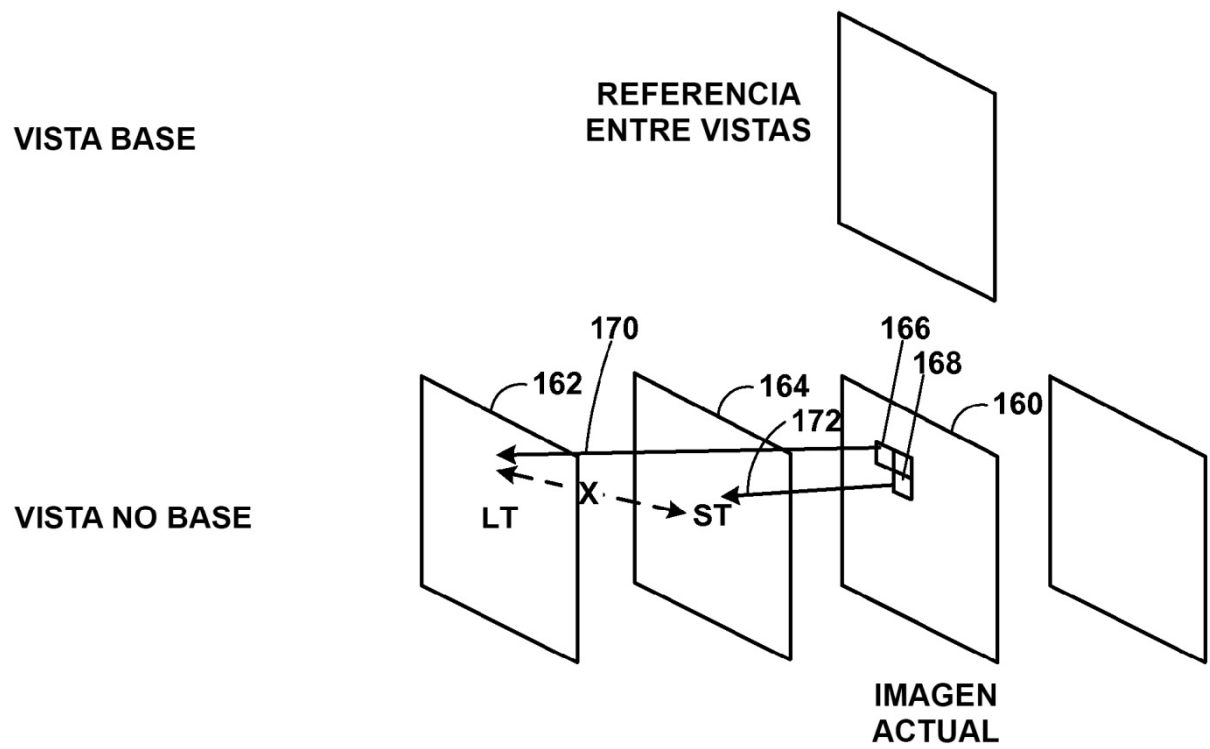


FIG. 8

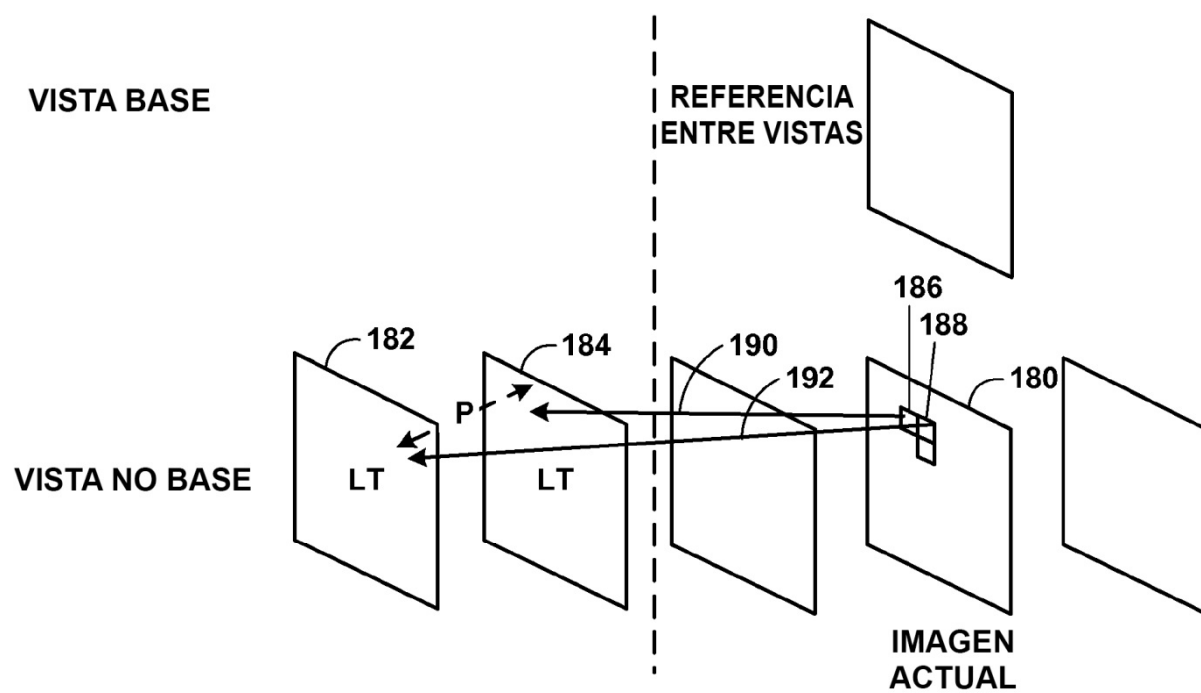
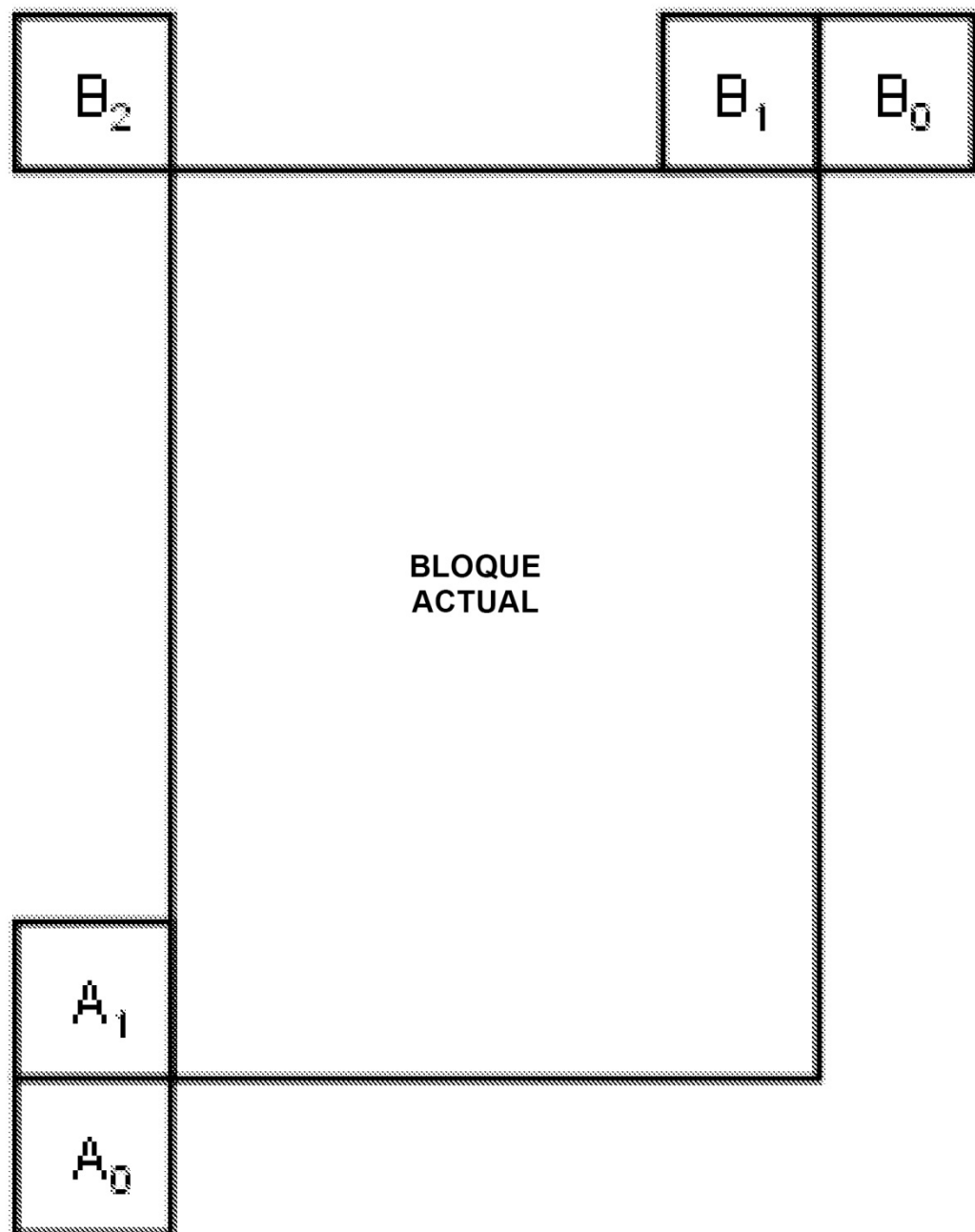


FIG. 9





**FIG. 10**

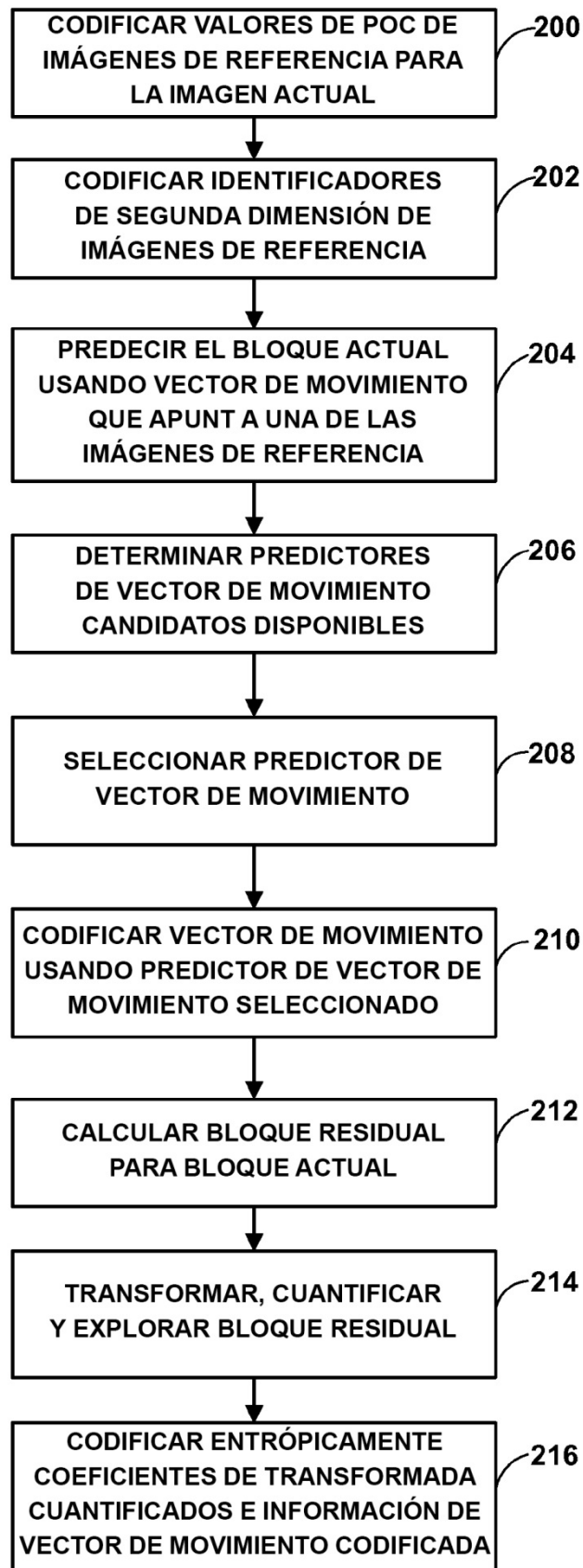


FIG. 11

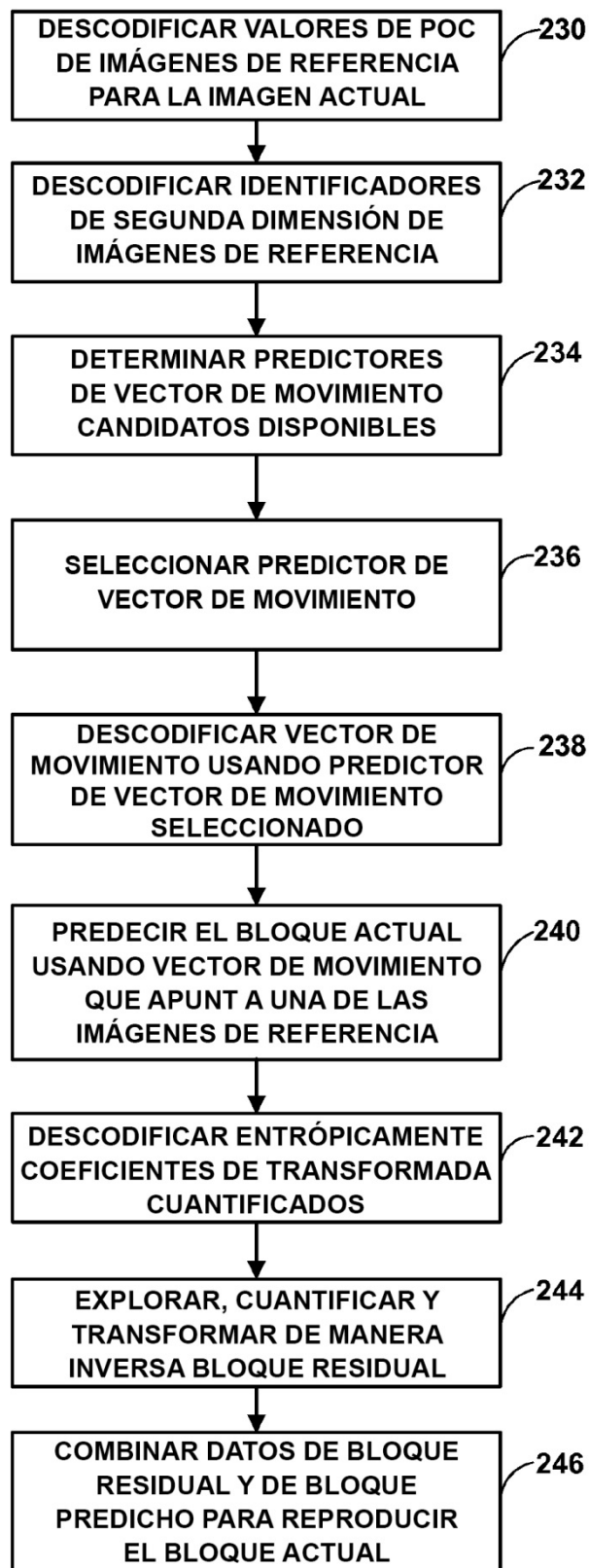


FIG. 12