

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 734 574**

51 Int. Cl.:

**H04N 19/52** (2014.01)

**H04N 19/597** (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **06.12.2013 PCT/US2013/073649**

87 Fecha y número de publicación internacional: **12.06.2014 WO14089475**

96 Fecha de presentación y número de la solicitud europea: **06.12.2013 E 13811104 (2)**

97 Fecha y número de publicación de la concesión europea: **10.04.2019 EP 2929688**

54 Título: **Modo avanzado de fusión/salto y modo avanzado de predicción de vector de movimiento (AMVP) para vídeo 3D**

30 Prioridad:

**07.12.2012 US 201261734929 P**

**30.12.2012 US 201261747350 P**

**05.12.2013 US 201314098151**

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

**10.12.2019**

73 Titular/es:

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

72 Inventor/es:

**ZHANG, LI;  
CHEN, YING y  
KARCZEWICZ, MARTA**

74 Agente/Representante:

**FORTEA LAGUNA, Juan José**

ES 2 734 574 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

Modo avanzado de fusión/salto y modo avanzado de predicción de vector de movimiento (AMVP) para vídeo 3D

5 **CAMPO TÉCNICO**

[0001] Esta divulgación se refiere a la codificación de vídeo.

10 **ANTECEDENTES**

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

25 [0003] Las técnicas de compresión de vídeo realizan predicción espacial (intra-imagen) y/o predicción temporal (entre  
 imágenes) para reducir o eliminar la redundancia intrínseca a las secuencias de vídeo. Para la codificación de vídeo  
 basada en bloques, un fragmento de vídeo (por ejemplo, una trama de vídeo o una parte de una trama de vídeo)  
 puede dividirse en bloques de vídeo, que también pueden denominarse bloques arbolados, unidades de codificación  
 (CU) y/o nodos de codificación. Los bloques de vídeo en un fragmento con intracodificación (I) de una imagen se  
 30 codifican usando predicción espacial con respecto a muestras de referencia en bloques contiguos en la misma imagen.  
 Los bloques de vídeo en un fragmento intercodificado (P o B) de una imagen pueden usar la predicción espacial con  
 respecto a muestras de referencia en bloques contiguos en la misma imagen o la predicción temporal con respecto a  
 muestras de referencia en otras imágenes de referencia. Las imágenes pueden denominarse tramas y las imágenes  
 de referencia pueden denominarse tramas de referencia.

35 [0004] La predicción espacial o temporal da como resultado un bloque predictivo para un bloque a codificar. Los datos  
 residuales representan diferencias de píxeles entre el bloque original que a codificar y el bloque predictivo. Un bloque  
 intercodificado se codifica de acuerdo a 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 a un modo de intracodificación y a los datos residuales.  
 40 Para una mayor compresión, los datos residuales pueden transformarse desde el dominio del píxel a un dominio de  
 transformada, dando como resultado coeficientes de transformada residuales, que a continuación se pueden  
 cuantificar. Los coeficientes de transformada cuantificados, dispuestos inicialmente en una formación bidimensional,  
 pueden explorarse con el fin de generar un vector unidimensional de coeficientes de transformada, y puede aplicarse  
 codificación por entropía para lograr aún más compresión.

45 [0005] La contribución "Merge candidates derivation from disparity vector [Obtención de candidatos de fusión a partir  
 de un vector de disparidad]" (ZHANG L ET AL., 10 de octubre de 2012, JCT3V-B0048) aborda la predicción de  
 movimiento para la codificación de vídeo 3D y de múltiples visualizaciones. Se divulga un procedimiento para obtener  
 candidatos de vectores de movimiento predichos mediante visualización basándose en bloques identificados por un  
 vector de disparidad que se desplaza por una desviación.

50 **SUMARIO**

[0006] En general, esta divulgación describe técnicas para mejorar la eficiencia de codificación de la predicción de  
 movimiento en codificación de vídeo 3D y de múltiples visualizaciones. Por ejemplo, las técnicas pueden utilizar  
 55 candidatos de vectores de movimiento predichos entre visualizaciones y entrevistar candidatos de vectores de  
 movimiento de disparidad como candidatos en una lista de candidatos para el modo de fusión/salto y el modo de  
 predicción de vector de movimiento avanzado (AMVP). Las técnicas pueden desplazar un vector de disparidad, para  
 un bloque actual se con predicción inter, mediante un primer desplazamiento para identificar un bloque de referencia  
 en una visualización de referencia para obtener el candidato de vector de movimiento de predicción entre  
 60 visualizaciones y desplazar el vector de disparidad un segundo, diferente desviación para convertir el vector de  
 disparidad desviado en el candidato de vector de movimiento de disparidad de entrevista.

[0007] En algunos ejemplos, las técnicas pueden incluir solo el candidato de vector de movimiento predicho entre  
 65 visualizaciones en la lista de candidatos si el candidato de vector de movimiento predicho entre visualizaciones está  
 disponible, e incluir solo el candidato de vector de movimiento de disparidad entre visualizaciones en la lista de  
 candidatos si el candidato de vector de movimiento predicho entre visualizaciones no está disponible. En algunos

ejemplos, es posible que las técnicas no incluyan el candidato de vector de movimiento de predicción entre visualizaciones en la lista de candidatos, incluso si está disponible, si la información de movimiento para otro candidato de vector de movimiento de predicción entre visualizaciones (por ejemplo, una obtenida sin desplazar el vector de disparidad) ya está incluida en la lista de candidatos.

5  
**[0008]** En un ejemplo, la divulgación describe un procedimiento para codificar datos de vídeo, con el procedimiento que incluye determinar si un bloque actual tiene predicción inter en el modo de fusión, modo de salto o modo de predicción de vector de movimiento avanzado (AMVP), si el bloque actual tiene predicción inter en modo AMVP, determinar si se debe obtener un candidato de vector de movimiento de disparidad entre visualizaciones (IDMVC) o un candidato de vector de movimiento predicho entre visualizaciones (IPMVC) basándose en al menos un índice de imagen de referencia, y si el bloque actual tiene predicción inter en modo de fusión o modo de salto, determinando la disponibilidad del IPMVC. En este ejemplo, el IPMVC comprende información de movimiento para un bloque de referencia en una visualización de referencia a la que hace referencia un vector de disparidad, para el bloque actual, desplazado un primer valor de desviación, y el IDMVC comprende el vector de disparidad, para el bloque actual, desplazado un segundo valor de desviación diferente. El procedimiento también incluye la determinación de si el IPMVC o el IDMVC se incluirán en una lista de candidatos, y la codificación mediante predicción inter del bloque actual en una imagen actual en una visualización actual, diferente a la visualización de referencia, basándose en la lista de candidatos.

20  
**[0009]** En un ejemplo, la divulgación describió un dispositivo para codificar datos de vídeo, con el dispositivo que comprende un codificador de vídeo configurado para determinar si un bloque actual tiene predicción inter en el modo de fusión, modo de salto o modo de predicción de vector de movimiento avanzado (AMVP), si el bloque actual tiene predicción inter en modo AMVP, determinar si obtener un candidato de vector de movimiento de disparidad de entrevista (IDMVC) o un candidato de vector de movimiento predicho entre visualizaciones (IPMVC) basándose en al menos un índice de imagen de referencia, y si el bloque actual tiene predicción inter en el modo de fusión o en el modo de salto, determinar la disponibilidad del IPMVC. En este ejemplo, el IPMVC comprende información de movimiento para un bloque de referencia en una visualización de referencia a la que hace referencia un vector de disparidad, para el bloque actual, desplazado un primer valor de desviación, y el IDMVC comprende el vector de disparidad, para el bloque actual, desplazado un segundo valor de desviación diferente. El codificador de vídeo también está configurado para determinar si el IPMVC o el IDMVC se incluirán en una lista de candidatos, y el código de predicción inter del bloque actual en una imagen actual en una visualización actual, diferente a la visualización de referencia, basándose en la lista de candidatos.

35  
**[0010]** En un ejemplo, la divulgación describe un medio de almacenamiento legible por ordenador que tiene instrucciones almacenadas en el mismo que cuando se ejecutan hacen que uno o más procesadores de un dispositivo para codificar datos de vídeo determine si un bloque actual tiene predicción inter en el modo de fusión, modo de salto o modo de predicción de vector de movimiento avanzado (AMVP), si el bloque actual tiene predicción inter en el modo AMVP, determinar si se debe obtener un candidato de vector de movimiento de disparidad entre visualizaciones (IDMVC) o un candidato de vector de movimiento predicho entre visualizaciones (IPMVC) basándose en al menos un índice de imagen de referencia, y si el bloque actual tiene predicción inter en el modo de fusión o en el modo de salto, determinar la disponibilidad del IPMVC. En este ejemplo, el IPMVC comprende información de movimiento para un bloque de referencia en una visualización de referencia a la que hace referencia un vector de disparidad, para el bloque actual, desplazado un primer valor de desviación, y el IDMVC comprende el vector de disparidad, para el bloque actual, desplazado un segundo valor de desviación diferente. Las instrucciones también hacen que uno o más procesadores determinen si el IPMVC o el IDMVC se incluirán en una lista de candidatos, y codifiquen mediante predicción inter el bloque actual en una imagen actual en una visualización actual, diferente a la visualización de referencia, basándose en la lista de candidatos.

50  
**[0011]** En un ejemplo, la divulgación describe un dispositivo para codificar datos de vídeo que comprende medios para determinar si un bloque actual tiene predicción inter en el modo de fusión, modo de salto o modo de predicción de vector de movimiento avanzado (AMVP), si el bloque actual tiene predicción inter en el modo AMVP, medios para determinar si se debe obtener un candidato de vector de movimiento de disparidad entre visualizaciones (IDMVC) o un candidato de vector de movimiento predicho entre visualizaciones (IPMVC) basado en al menos un índice de imagen de referencia, y si el bloque actual tiene predicción inter en modo de fusión o modo de salto, medios para determinar la disponibilidad del IPMVC. En este ejemplo, el IPMVC comprende información de movimiento para un bloque de referencia en una visualización de referencia a la que hace referencia un vector de disparidad, para el bloque actual, desplazado un primer valor de desviación, y el IDMVC comprende el vector de disparidad, para el bloque actual, desplazado un segundo valor de desviación diferente. El dispositivo también incluye medios para determinar si el IPMVC o el IDMVC se incluirán en una lista de candidatos, y medios para la codificación mediante predicción inter del bloque actual en una imagen actual en una visualización actual, diferente a la visualización de referencia, basándose en la lista de candidatos.

65  
**[0012]** Los detalles de uno o más ejemplos se exponen en los dibujos adjuntos y en la siguiente descripción. Otras características, objetos y ventajas resultarán evidentes a partir de la descripción y de los dibujos, y a partir de las reivindicaciones.

**BREVE DESCRIPCIÓN DE LOS DIBUJOS**

**[0013]**

- 5 La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y descodificación de vídeo a modo de ejemplo que puede utilizar las técnicas de la presente divulgación.
- La FIG. 2 es un diagrama conceptual que ilustra un ejemplo de orden de descodificación para vídeo de múltiples visualizaciones.
- 10 La FIG. 3 es un diagrama conceptual que ilustra un ejemplo de estructura de predicción para vídeo de múltiples visualizaciones.
- La FIG. 4 muestra un conjunto de ejemplo de bloques candidatos que se pueden usar tanto en modo de fusión como en modo AMVP.
- 15 La FIG. 5 muestra un ejemplo de bloques que se comprueban en una imagen candidata para determinar un vector de disparidad para un bloque actual.
- La FIG. 6 es un diagrama conceptual que ilustra un proceso de obtención de ejemplo de un candidato de vector de movimiento predicho entre visualizaciones.
- 20 La FIG. 7 es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo que puede implementar las técnicas de predicción inter de esta divulgación.
- 25 La FIG. 8 es un diagrama de bloques que ilustra un ejemplo de descodificador de vídeo que puede implementar las técnicas de predicción inter de esta divulgación.
- La FIG. 9 es un diagrama de flujo que muestra un proceso de codificación a modo de ejemplo de acuerdo con las técnicas de la divulgación.
- 30 La FIG. 10 es un diagrama de flujo que muestra un proceso de descodificación de ejemplo de acuerdo con las técnicas de la divulgación.

**35 DESCRIPCIÓN DETALLADA**

- 40 **[0014]** Esta divulgación describe técnicas relacionadas con la señalización de parámetros de movimiento en la codificación de vídeo de múltiples visualizaciones, incluso en la nueva norma de codificación de vídeo de alta eficiencia tridimensional, comúnmente denominada HEVC 3D. Las técnicas de compresión de vídeo pueden incluir la predicción de un bloque actual de datos de vídeo basado en un bloque de datos de vídeo ya codificado. En esta divulgación, el término bloque actual se refiere en general a un bloque de datos de vídeo que se codifican actualmente (por ejemplo, codifican o descodifican), en oposición a los bloques de vídeo que ya están codificados o los bloques de vídeo que aún deben codificarse. En un modo de predicción intra, el bloque actual se predice basándose en uno o más bloques previamente codificados en la misma imagen que el bloque actual, mientras que en un modo de predicción inter se predice el bloque actual basándose en un bloque ya codificado en una imagen diferente. La imagen diferente puede ser una imagen de una instancia de tiempo diferente desde la misma visualización que el bloque actual (por ejemplo, una imagen en otra unidad de acceso que la unidad de acceso que incluye la imagen actual) o puede ser una imagen de la misma instancia de tiempo, pero desde una visualización diferente (por ejemplo, una imagen en la misma unidad de acceso que la unidad de acceso que incluye la imagen actual).
- 50 **[0015]** Cuando un descodificador de vídeo debe realizar una predicción inter para un bloque de vídeo dado, los parámetros de movimiento se señalan en un flujo de bits de vídeo codificado. Los parámetros de movimiento definen, por ejemplo, un vector de movimiento, un índice de imagen de referencia y una dirección de predicción para que el descodificador de vídeo pueda determinar qué bloque en qué imagen usar como bloque predictivo para el bloque actual. Un codificador de vídeo puede usar varios modos para señalar parámetros de movimiento en un flujo de bits codificado. Con el fin de reducir el número de bits necesarios para implementar la codificación con predicción inter, algunos de esos modos pueden utilizar la predicción de los parámetros de movimiento en lugar de la señalización explícita de los parámetros de movimiento. En el desarrollo de la norma de codificación de vídeo de alta eficiencia (HEVC), existen varios modos para la predicción de parámetros de movimiento, uno de los cuales se conoce como modo de fusión. En el modo de fusión, un codificador de vídeo construye una lista de candidatos de parámetros de movimiento (por ejemplo, imágenes de referencia y vectores de movimiento) utilizando como candidatos parámetros de movimiento de bloques contiguos, incluidos bloques contiguos espaciales y bloques contiguos temporales, y en el caso de 3D-HEVC también incluyendo bloques contiguos entre visualizaciones. Los parámetros de movimiento seleccionados se señalan desde un codificador de vídeo a un descodificador de vídeo transmitiendo un índice del candidato seleccionado de la lista de candidatos. En el descodificador de vídeo, una vez que el índice es descodificado, todos los parámetros de movimiento del bloque correspondiente del candidato seleccionado son

heredados. Un codificador de vídeo y un decodificador de vídeo están configurados para construir las mismas listas basándose en bloques ya codificados. Por lo tanto, basándose en el índice, el decodificador de vídeo puede identificar los parámetros de movimiento del candidato seleccionado por el codificador de vídeo.

5 **[0016]** Otro modo para predecir los parámetros de movimiento es el modo de salto. El modo de salto en general funciona de la misma manera descrita anteriormente para el modo de fusión, pero en el modo de salto, no se agregan datos residuales al bloque predicho, mientras que en el modo de fusión, los datos residuales se agregan al bloque predicho. La construcción de una lista y la transmisión de un índice para identificar a un candidato en la lista descrita anteriormente con referencia al modo de fusión en general también se realiza en modo de salto. Algunos códecs de vídeo pueden tratar el modo de salto como un modo separado del modo de fusión, mientras que otras normas de codificación de vídeo, incluyendo HEVC y sus extensiones, pueden tratar el modo de salto como un caso especial del modo de fusión.

15 **[0017]** Otro modo para predecir los parámetros de movimiento es el modo avanzado de predicción de vector de movimiento (AMVP). En el modo AMVP, se obtiene una lista de candidatos de predictores de vector de movimiento para cada hipótesis de movimiento basándose en el índice de referencia codificado. Esta lista incluye vectores de movimiento de bloques contiguos que están asociados con el mismo índice de referencia, así como un predictor de vector de movimiento temporal que se obtiene basándose en los parámetros de movimiento del bloque contiguo del bloque que se encuentra en el mismo lugar en una imagen de referencia temporal. Los vectores de movimiento elegidos se señalan transmitiendo un índice a la lista de candidatos. Además, también se señalan los valores del índice de referencia y las diferencias del vector de movimiento. En esta divulgación, el término predictor de vector de movimiento se usa en general para referirse a cualquier vector de movimiento a partir del cual se predicen uno o más vectores de movimiento. En algunos casos, el predictor de vector de movimiento y el vector de movimiento predicho pueden ser idénticos, mientras que en otros casos el predictor de vector de movimiento y el vector de movimiento predicho pueden ser diferentes. En el modo AMVP, por ejemplo, el vector de movimiento predicho corresponde al predictor del vector de movimiento más los valores de diferencia del vector de movimiento. Esta divulgación también se refiere al término predictor de movimiento direccional, que en general se refiere a un predictor de vector de movimiento asociado con una dirección particular (es decir, una lista de imágenes de referencia particular). En el caso de la bi-predicción, un predictor de vector de movimiento puede incluir dos predictores de movimiento direccional.

30 **[0018]** Un vector de disparidad se usa en general para identificar la ubicación de un bloque en una imagen de referencia entre visualizaciones que corresponde a un bloque actual de una visualización actual. Para obtener un vector de disparidad (DV) para un bloque actual, se utiliza una técnica denominada obtención de vector de disparidad basada en bloques contiguos (NBDV) en algunas versiones actuales del modelo de prueba 3D-HEVC (3D-HTM). NBDV utiliza vectores de movimiento de disparidad de bloques contiguos espaciales y temporales del bloque actual. En NBDV, los vectores de movimiento de disparidad de los bloques contiguos espaciales o temporales se controlan en un orden fijo. Una vez que se identifica un vector de movimiento de disparidad (es decir, un vector de movimiento que apunta a una imagen de referencia entre visualizaciones), el proceso de comprobación finaliza y el vector de movimiento de disparidad identificado se convierte a DV para el bloque actual. El DV se puede utilizar en predicción de movimiento entre visualizaciones y/o predicción de residuos entre visualizaciones. Si no se encuentra un vector de movimiento de disparidad después de comprobar todos los bloques contiguos predefinidos, se puede utilizar un DV cero para la predicción de movimiento entre visualizaciones mientras que la predicción residual entre visualizaciones puede estar deshabilitada para la unidad de predicción (PU) correspondiente.

45 **[0019]** En algunos ejemplos, el vector de disparidad para el bloque actual puede obtenerse de un vector de disparidad de un bloque contiguo. Por ejemplo, un bloque contiguo (por ejemplo, un bloque contiguo espacial o temporal) puede no haber tenido predicción inter con un vector de movimiento de disparidad. Sin embargo, durante la codificación o decodificación del bloque contiguo, se puede haber obtenido un vector de disparidad para ese bloque contiguo. Este vector de disparidad obtenido a partir del bloque contiguo puede usarse como el vector de disparidad para el bloque actual. Por ejemplo, si ninguno de los bloques contiguos, que se evalúan, tiene predicción inter con un vector de movimiento de disparidad, pero un vector de disparidad obtenido sale para un bloque contiguo, un codificador de vídeo y un decodificador de vídeo pueden establecer el vector de disparidad obtenido a partir del bloque contiguo como vector de disparidad para el bloque actual.

55 **[0020]** Basándose en el DV, se puede agregar un nuevo vector de movimiento candidato, denominado vector de movimiento predicho entre visualizaciones, si está disponible, a las listas de candidatos para los modos AMVP y de salto/fusión. El vector de movimiento predicho entre visualizaciones, si está disponible, es un vector de movimiento temporal. En general, esta divulgación describe técnicas para mejorar la eficiencia de codificación de la predicción de movimiento en codificación de vídeo 3D y de múltiples visualizaciones.

60 **[0021]** La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y decodificación de vídeo 10 de ejemplo que puede utilizar técnicas de esta divulgación. Como se muestra en la FIG. 1, el sistema 10 incluye un dispositivo de origen 12 que proporciona datos de vídeo codificados, a decodificar más adelante por un dispositivo de destino 14. En particular, el dispositivo de origen 12 proporciona los datos de vídeo al dispositivo de destino 14 por medio de un medio legible por ordenador 16. El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender cualquiera entre una amplia gama de dispositivos, incluidos ordenadores de escritorio, ordenadores

plegables (es decir, portátiles), ordenadores tipo tablet, descodificadores, equipos telefónicos de mano tales como los denominados teléfonos «inteligentes», los denominados paneles «inteligentes», televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de transmisión de vídeo en tiempo real o similares. En algunos casos, el dispositivo de origen 12 y el dispositivo de destino 14 pueden estar equipados para la comunicación inalámbrica.

**[0022]** El dispositivo de destino 14 puede recibir los datos de vídeo codificados, a descodificar mediante el medio legible por ordenador 16. El medio legible por ordenador 16 puede comprender cualquier tipo de medio o dispositivo capaz de desplazar los datos de vídeo codificados desde el dispositivo de origen 12 al dispositivo de destino 14. En un ejemplo, el medio legible por ordenador 16 puede comprender un medio de comunicación para permitir al dispositivo de origen 12 transmitir datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real. Los datos de vídeo codificados se pueden modular de acuerdo a 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 cableada, tal como un espectro de radiofrecuencia (RF) o una o más líneas físicas de transmisión. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área amplia o una red global tal como Internet. El medio de comunicación puede incluir routers, 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.

**[0023]** En algunos ejemplos, pueden emitirse datos codificados desde la interfaz de salida 22 hasta un dispositivo de almacenamiento. De forma similar, se puede acceder a los datos codificados del dispositivo de almacenamiento mediante una interfaz de entrada 28. El dispositivo de almacenamiento puede incluir cualquiera entre varios medios de almacenamiento de datos, distribuidos o de acceso local, tales como un disco duro, discos Blu-ray, DVD, CD-ROM, memoria flash, memoria volátil o no volátil, o cualquier otro medio de almacenamiento digital adecuado para almacenar datos de vídeo codificados. En un ejemplo adicional, el dispositivo de almacenamiento puede corresponder a un servidor de ficheros o a otro dispositivo de almacenamiento intermedio que pueda almacenar el vídeo codificado generado por el dispositivo de origen 12. El dispositivo de destino 14 puede acceder a datos de vídeo almacenados desde el dispositivo de almacenamiento, mediante transmisión continua o descarga. El servidor de ficheros puede ser cualquier tipo de servidor capaz de almacenar datos de vídeo codificados y transmitir esos datos de vídeo codificados al dispositivo de destino 14. Los servidores de ficheros de ejemplo incluyen un servidor de la Red (por ejemplo, para una sede de la Red), un servidor del FTP, dispositivos de almacenamiento conectados en red (NAS) o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de cualquier conexión de datos estándar, incluida una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión de Wi-Fi), una conexión cableada (por ejemplo, DSL, módem de cable, etc.) o una combinación de ambas que sea adecuada para acceder a datos de vídeo codificados almacenados en un servidor de ficheros. La transmisión de datos de vídeo codificados desde el dispositivo de almacenamiento puede ser una transmisión continua, una transmisión de descarga o una combinación de ambas.

**[0024]** Las técnicas de esta divulgación no están limitadas necesariamente a aplicaciones o configuraciones inalámbricas. Las técnicas se pueden aplicar a la codificación de vídeo como soporte de cualquiera entre varias aplicaciones de multimedia, tales como difusiones de televisión por el aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones continuas de vídeo por Internet, tales como la transmisión continua adaptativa dinámica por HTTP (DASH), el vídeo digital que se codifica en un medio de almacenamiento de datos, la descodificación de vídeo digital almacenado en un medio de almacenamiento de datos u otras aplicaciones. En algunos ejemplos, el sistema 10 puede estar configurado para prestar soporte a una transmisión de vídeo unidireccional o bidireccional, a fin de prestar soporte a aplicaciones tales como la transmisión de vídeo en tiempo real, la reproducción de vídeo, la difusión de vídeo y/o la videotelefonía.

**[0025]** En el ejemplo de la FIG. 1, el dispositivo de origen 12 incluye un origen de vídeo 18, una unidad de estimación de profundidad 19, un codificador de vídeo 20 y una interfaz de salida 22. El dispositivo de destino 14 incluye la interfaz de entrada 28, el descodificador de vídeo 30, la unidad de representación basada en la imagen de profundidad (DIBR) 31 y el dispositivo de visualización 32. En otros ejemplos, un dispositivo de origen y un dispositivo de destino pueden incluir otros componentes o disposiciones. Por ejemplo, el dispositivo de origen 12 puede recibir datos de vídeo desde un origen de vídeo externo 18, tal como una cámara externa. Del mismo modo, el dispositivo de destino 14 puede interactuar con un dispositivo de visualización externo, en lugar de incluir un dispositivo de visualización integrado. Muchas de las técnicas descritas en esta divulgación pueden realizarse tanto por el codificador de vídeo 20 como por el descodificador de vídeo 30. Por lo tanto, para facilitar la explicación, el codificador de vídeo 20 y el descodificador de vídeo 30 a veces pueden denominarse conjuntamente codificador de vídeo 20/30. Las técnicas descritas con relación al codificador de vídeo 20/30 pueden realizarse ya sea mediante el codificador de vídeo 20 o el descodificador de vídeo 30.

**[0026]** El sistema ilustrado 10 de la FIG. 1 es simplemente un ejemplo. Las técnicas de esta divulgación pueden realizarse mediante cualquier dispositivo de codificación y/o descodificación de vídeo digital. Aunque, en general, las técnicas de la presente divulgación son llevadas a cabo por un dispositivo de codificación de vídeo, las técnicas también pueden ser llevadas a cabo por un codificador/descodificador de vídeo, denominado típicamente "CÓDEC". Además, las técnicas de la presente divulgación también se pueden llevar a cabo mediante un preprocesador de

vídeo. El dispositivo de origen 12 y el dispositivo de destino 14 son simplemente ejemplos de dichos dispositivos de codificación en los que el dispositivo de origen 12 genera datos de vídeo codificados para su transmisión al dispositivo de destino 14. En algunos ejemplos, los dispositivos 12, 14 pueden funcionar de manera esencialmente simétrica, de modo que cada uno de los dispositivos 12, 14 incluya componentes de codificación y descodificación de vídeo. De ahí que el sistema 10 pueda prestar soporte a la transmisión de vídeo unidireccional o bidireccional entre los dispositivos de vídeo 12, 14, por ejemplo, para la transmisión continua de vídeo, la reproducción de vídeo, la difusión de vídeo o la videotelefonía.

**[0027]** El origen de vídeo 18 del dispositivo de origen 12 puede incluir un dispositivo de captura de vídeo, tal como una cámara de vídeo, un archivo de vídeo que contiene vídeo capturado previamente y/o una interfaz de suministro de vídeo para recibir vídeo desde un proveedor de contenidos de vídeo. Como otra alternativa, el origen de vídeo 18 puede generar datos basados en gráficos por ordenador como el vídeo de origen, o una combinación de vídeo en directo, vídeo archivado y vídeo generado por ordenador. En algunos casos, si el origen de vídeo 18 es una videocámara, el dispositivo de origen 12 y el dispositivo de destino 14 pueden formar los llamados teléfonos con cámara o videoteléfonos. Sin embargo, como se ha mencionado anteriormente, las técnicas descritas en la presente divulgación pueden ser aplicables a la codificación de vídeo en general, y se pueden aplicar a aplicaciones inalámbricas y/o cableadas. En cada caso, el codificador de vídeo 20 puede codificar el vídeo capturado, precapturado o generado por ordenador. La información de vídeo codificada se puede emitir, a continuación, por la interfaz de salida 22 hacia un medio legible por ordenador 16.

**[0028]** El origen de vídeo 18 puede proporcionar múltiples visualizaciones de datos de vídeo al codificador de vídeo 20. Por ejemplo, el origen de vídeo 18 puede corresponder a una formación de cámaras, teniendo cada una una posición horizontal única con respecto a una escena particular que se está filmando. De forma alternativa, el origen de vídeo 18 puede generar datos de vídeo desde perspectivas dispares de cámara horizontal, por ejemplo, usando gráficos por ordenador. La unidad de estimación de profundidad 19 puede configurarse para determinar valores para píxeles de profundidad correspondientes a píxeles en una imagen de textura. Por ejemplo, la unidad de estimación de profundidad 19 puede representar una unidad de Navegación y Medición por Sonido (SONAR), una unidad de Detección y Medición de Luz (LIDAR) u otra unidad capaz de determinar directamente los valores de profundidad de forma esencialmente simultánea a la grabación de datos de vídeo de una escena.

**[0029]** De forma adicional o alternativa, la unidad de estimación de profundidad 19 puede configurarse para calcular valores de profundidad de manera indirecta mediante la comparación de dos o más imágenes que se capturaron esencialmente al mismo tiempo desde diferentes perspectivas de cámara horizontal. Calculando la disparidad horizontal entre valores de píxeles esencialmente similares en las imágenes, la unidad de estimación de profundidad 19 puede aproximarse a la profundidad de varios objetos en la escena. La unidad de estimación de profundidad 19 puede estar integrada funcionalmente con el origen de vídeo 18, en algunos ejemplos. Por ejemplo, cuando el origen de vídeo 18 genera imágenes de gráficos por ordenador, la unidad de estimación de profundidad 19 puede proporcionar mapas de profundidad reales para objetos gráficos, por ejemplo, usando coordenadas z de píxeles y objetos usados para representar imágenes de textura.

**[0030]** El medio legible por ordenador 16 puede incluir medios transitorios, tales como una difusión inalámbrica o una transmisión de red cableada, o medios de almacenamiento (es decir, medios de almacenamiento no transitorio), tales como un disco duro, una unidad de memoria flash, un disco compacto, un disco de vídeo digital, un disco Blu-ray u otros medios legibles por ordenador. En algunos ejemplos, un servidor de red (no mostrado) puede recibir datos de vídeo codificados desde el dispositivo de origen 12 y proporcionar los datos de vídeo codificados al dispositivo de destino 14, por ejemplo, mediante transmisión por red. De forma similar, un dispositivo informático de una utilidad de producción de un medio, tal como una utilidad de grabación 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.

**[0031]** La interfaz de entrada 28 del dispositivo de destino 14 recibe información desde el medio legible por ordenador 16. La información del medio legible por ordenador 16 puede incluir información sintáctica definida por el codificador de vídeo 20, que también es utilizada por el descodificador de vídeo 30, que incluye elementos sintácticos que describen características y/o procesamiento de bloques y otras unidades codificadas, por ejemplo, grupos de imágenes (GOP). El dispositivo de visualización 32 muestra los datos de vídeo descodificados a un usuario, y puede comprender cualquiera entre varios dispositivos de visualización, tales como un tubo de rayos catódicos (CRT), una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de visualización. En algunos ejemplos, el dispositivo de visualización 32 puede comprender un dispositivo capaz de mostrar dos o más visualizaciones de forma simultánea o esencialmente simultánea, por ejemplo, para producir un efecto visual 3D para un espectador.

**[0032]** La unidad de DIBR 31 del dispositivo de destino 14 puede representar visualizaciones sintetizadas usando información de textura y profundidad de visualizaciones descodificadas, recibidas desde el descodificador de vídeo 30. Por ejemplo, la unidad de DIBR 31 puede determinar la disparidad horizontal para datos de píxeles de imágenes de textura como una función de los valores de píxeles en mapas de profundidad correspondientes. La unidad de DIBR

31 puede generar entonces una imagen sintetizada mediante el desplazamiento de píxeles en una imagen de textura, a la izquierda o a la derecha, en la disparidad horizontal determinada. De esta manera, el dispositivo de visualización 32 puede mostrar una o más visualizaciones, que pueden corresponder a visualizaciones descodificadas y/o visualizaciones sintetizadas, en cualquier combinación. De acuerdo con las técnicas de esta divulgación, el descodificador de vídeo 30 puede proporcionar valores de precisión originales y actualizados para los rangos de profundidad y parámetros de la cámara a la unidad DIBR 31, que puede usar los rangos de profundidad y los parámetros de la cámara para sintetizar adecuadamente las visualizaciones.

**[0033]** Aunque no se muestra en la FIG. 1, en algunos aspectos, tanto el codificador de vídeo 20 como el descodificador de vídeo 30 pueden estar integrados con un codificador y descodificador de audio, y pueden incluir unidades MUX-DEMUX adecuadas, u otro tipo de hardware y software, para ocuparse de la codificación tanto de audio como de vídeo en un flujo de datos común o en flujos de datos separados. Si procede, las unidades MUX-DEMUX se pueden ajustar al protocolo multiplexador H.223 de la ITU o a otros protocolos tales como el protocolo de datagramas de usuario (UDP).

**[0034]** 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 eficiencia (HEVC), y pueden ajustarse al modelo de prueba de HEVC (HM). Un borrador reciente de la próxima norma HEVC se describe en el documento HCTVC-J1003, de Bross, et al., titulado "High Efficiency Video Coding (HEVC) Text Specification Draft 8 [Borrador de Memoria Descriptiva Textual de Codificación de Vídeo de Alta Eficiencia 8]", Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 10.<sup>a</sup> reunión: Estocolmo, Suecia, del 11 de julio de 2012 al 12 de julio de 2012, que, desde el 4 de diciembre de 2013, puede descargarse en [http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/10\\_Stockholm/wg11/JCTVC-J1003-v8.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/10_Stockholm/wg11/JCTVC-J1003-v8.zip). Otro borrador reciente de la norma HEVC, denominado "HEVC Working Draft 10 [Borrador 10 de trabajo de la HEVC]" o "WD10", se describe en el documento JCTVC-L1003v34, de Bross et al., titulado "High Efficiency video Coding (HEVC) Text Specification Draft 10 (for FDIS & Last Call)" ["Memoria descriptiva textual de la Codificación de Vídeo de Alta Eficacia (HEVC), Borrador 10 (para Norma Internacional de Borrador Final y Última Llamada)"], Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 12.<sup>a</sup> reunión: Ginebra, Suiza, del 14 al 23 de enero de 2013, que, a partir del 4 de diciembre de 2013, puede descargarse en [http://phenix.intevry.fr/jct/doc\\_end\\_user/documents/12\\_Geneva/wg11/JCTVC-L1003-v34.zip](http://phenix.intevry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v34.zip). Otro borrador más de la norma HEVC, se denomina en el presente documento "WD10 revisions [Revisiones de WD10]", descrito en el artículo de Bross et al., "Editors' proposed corrections to HEVC version 1 [Correcciones propuestas de los editores para la HEVC, versión 1]", Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 13.<sup>a</sup> reunión, Incheon, Corea, abril de 2013, que, a partir del 4 de diciembre de 2013, está disponible en: [http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/13\\_Incheon/wg11/JCTVC-M0432-v3.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/13_Incheon/wg11/JCTVC-M0432-v3.zip).

**[0035]** Con fines de ilustración, las técnicas de esta divulgación se describen principalmente con respecto a la ampliación 3DV de HEVC. Sin embargo, debe entenderse que estas técnicas pueden aplicarse a otras normas para codificar datos de vídeo utilizados para producir también un efecto tridimensional.

**[0036]** Un Equipo de Colaboración Conjunta en Codificación de Vídeo 3D (JCT-3C) de VCEG y MPEG está elaborando una norma 3DV basada en HEVC, para la cual una parte de los trabajos de normalización incluye la normalización del códec de vídeo de múltiples visualizaciones basado en HEVC (MV-HEVC) y otra parte, la codificación de vídeo 3D basada en HEVC (3D-HEVC). Para MV-HEVC, en general se desea que haya solo cambios sintácticos de alto nivel (HLS) en relación con HEVC, de modo que ningún módulo en el nivel CU/PU en HEVC necesite ser rediseñado y pueda reutilizarse completamente para MV-HEVC. Para 3D-HEVC, se pueden incluir y soportar nuevas herramientas de codificación, incluyendo aquellas en la codificación a nivel de unidad/unidad de predicción, tanto para las visualizaciones de textura como de profundidad. El software más actual 3D-HTM para 3D-HEVC se puede descargar, a partir del 4 de diciembre de 2013, en el siguiente enlace: [https://hevc.hhi.fraunhofer.de/svn/svn\\_HEVCSoftware/trunk/](https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/trunk/) or [https://hevc.hhi.fraunhofer.de/svn/svn\\_3DVCSoftware/tags/HTM-5.0/](https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-5.0/).

**[0037]** De forma alternativa, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo a otras normas privadas o industriales, tales como la norma ITU-T H.264, de forma alternativa denominada MPEG-4, Parte 10, codificación de vídeo avanzada (AVC), o ampliaciones de dichas normas, como la ampliación MVC de ITU-T H.264/AVC. En particular, las técnicas de esta divulgación se relacionan con la codificación de vídeo de múltiples visualizaciones y/o 3D basada en códecs avanzados. En general, las técnicas de esta divulgación se pueden aplicar a cualquiera de una variedad de normas de codificación de vídeo diferentes. Por ejemplo, estas técnicas se pueden aplicar a la ampliación de codificación de vídeo de múltiples visualizaciones (MVC) de ITU-T H.264/AVC (codificación avanzada de vídeo), a una ampliación de vídeo 3D (3DV) de la próxima norma HEVC (por ejemplo, 3D-HEVC) u otra norma de codificación.

**[0038]** La norma ITU-T H.264/MPEG-4 (AVC) fue formulada por el Grupo de Expertos en Codificación de Vídeo (VCEG) de la ITU-T junto con el Grupo de Expertos en Imágenes en Movimiento (MPEG) de la ISO/IEC, como el producto de una asociación colectiva conocida como el Equipo Mixto de Vídeo (JVT, por sus siglas en inglés). En algunos aspectos, las técnicas descritas en la presente divulgación se pueden aplicar a dispositivos que se ajustan

en general a la norma H.264. La norma H.264 está descrita en la recomendación H.264 de ITU-T, codificación avanzada de vídeo para servicios audiovisuales genéricos, por el grupo de estudio de ITU-T, y con fecha de marzo de 2005, que se puede denominar en el presente documento 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 extensiones para la H.264/MPEG-4 AVC.

**[0039]** Tanto el codificador de vídeo 20 como el descodificador de vídeo 30 pueden implementarse como cualquiera entre varios circuitos codificadores adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), formaciones de puertas programables in situ (FPGA), lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. Cuando las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio adecuado no transitorio legible por ordenador, y ejecutar las instrucciones en hardware usando uno o más procesadores para realizar las técnicas de esta divulgación. Cada uno entre el codificador de vídeo 20 y el descodificador de vídeo 30 se puede incluir en uno o más codificadores o descodificadores, cualquiera de los cuales se puede integrar como parte de un codificador/descodificador (CÓDEC) combinado en un dispositivo respectivo. Un dispositivo que incluye el codificador de vídeo 20 y/o el descodificador de vídeo 30 puede comprender un circuito integrado, un microprocesador y/o un dispositivo de comunicación inalámbrica, tal como un teléfono celular.

**[0040]** Inicialmente, se analizarán las técnicas de codificación de ejemplo de la HEVC. El JCT-VC ha desarrollado la norma HEVC. La iniciativa de normalización HEVC se basa en un modelo en evolución de un dispositivo de codificación de vídeo denominado modelo de prueba HEVC (HM). El HM supone varias capacidades adicionales de los dispositivos de codificación de vídeo respecto a los dispositivos existentes de acuerdo con, por ejemplo, la norma ITU-T H.264/AVC. Por ejemplo, mientras que la norma H.264 proporciona nueve modos de codificación mediante predicción intra, el HM puede proporcionar hasta treinta y tres modos de codificación mediante predicción intra angulares más modos DC y planares.

**[0041]** En general, el modelo de explotación del HM especifica que una trama o imagen de vídeo puede dividirse en una secuencia de bloques de árbol o unidades de codificación más grandes (LCU), que incluyen tanto muestras de luma como de croma. Los datos sintácticos dentro de un flujo de bits pueden definir un tamaño para la LCU, que es una máxima unidad de codificación en lo que respecta al número de píxeles. Un fragmento incluye un número de bloques de árbol consecutivos en orden de codificación. Una trama o imagen de vídeo puede dividirse en uno o más fragmentos. Cada bloque arbolado puede dividirse en unidades de codificación (CU) de acuerdo a un árbol cuádruple. En general, una estructura de datos de árbol cuádruple incluye un nodo por CU, con un nodo raíz correspondiente al bloque arbolado. 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.

**[0042]** Cada nodo de la estructura de datos en árbol cuádruple puede proporcionar datos sintácticos para la CU correspondiente. Por ejemplo, un nodo en el árbol cuádruple puede incluir un indicador de división, que indica si la CU correspondiente al nodo está dividida en varias sub-CU. Los elementos sintácticos para una CU pueden definirse de manera recursiva y pueden depender de si la CU está dividida en varias sub-CU. Si una CU no está dividida adicionalmente, se denomina CU hoja. En esta divulgación, cuatro sub-CU de una CU hoja también se denominarán CU hojas incluso aunque no haya ninguna división explícita de la CU hoja original. Por ejemplo, si una CU con un tamaño de 16x16 no se divide adicionalmente, las cuatro sub-CU de tamaño 8x8 también se denominarán CU hojas aunque la CU de tamaño 16x16 no se haya dividido nunca.

**[0043]** Una CU tiene un propósito similar a un macrobloque de la norma H.264, excepto porque una CU no tiene una distinción de tamaño. Por ejemplo, un bloque arbolado puede dividirse en cuatro nodos secundarios (también denominados sub-CU) y cada nodo secundario puede a su vez ser un nodo principal y dividirse en otros cuatro nodos secundarios. Un nodo secundario final, no dividido, denominado nodo hoja del árbol cuádruple, comprende un nodo de codificación, también denominado CU hoja. Los datos sintácticos asociados a un flujo de bits codificado pueden definir un número máximo de veces en que puede dividirse un bloque arbolado, denominado profundidad de CU máxima, y también pueden definir un tamaño mínimo de los nodos de codificación. Por consiguiente, un flujo de bits también puede definir una unidad de codificación de tamaño mínimo (SCU). Esta divulgación utiliza el término "bloque" para referirse a cualquiera entre una CU, PU o TU, en el contexto de la HEVC, o a estructuras de datos similares en el contexto de otras normas (por ejemplo, macrobloques y subbloques de los mismos en la norma H.264/AVC).

**[0044]** 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 ser de forma cuadrada. El tamaño de la CU puede variar desde 8x8 píxeles hasta el tamaño del bloque arbolado, con un máximo de 64x64 píxeles o más. Cada CU puede contener una o más PU y una o más TU. Los datos sintácticos asociados a una CU pueden describir, por ejemplo, la división de la CU en una o más PU. Los modos de división pueden diferir en función de si la CU está codificada en modo de salto o fusión, codificada en modo de predicción intra o codificada en modo de predicción inter. Las PU pueden dividirse para tener forma no cuadrada. Los datos sintácticos asociados a una CU también pueden describir, por ejemplo, la división de la CU en una o más TU de acuerdo con un árbol cuádruple. Una TU puede tener forma cuadrada o no cuadrada (por ejemplo, rectangular).

**[0045]** La norma HEVC permite transformaciones de acuerdo con las TU, que pueden ser diferentes para diferentes CU. El tamaño de las TU se basa típicamente en el tamaño de las PU dentro de una CU definida dada para una LCU dividida, aunque puede que no sea siempre así. Las TU son típicamente del mismo tamaño o de un tamaño más pequeño que las PU. En algunos ejemplos, las muestras residuales correspondientes a una CU pueden subdividirse en unidades más pequeñas utilizando una estructura de árbol cuádruple conocida como «árbol cuádruple residual» (RQT). Los nodos hoja del RQT pueden denominarse unidades de transformada (TU). Los valores de diferencias de píxeles asociados a las TU pueden transformarse para generar coeficientes de transformada, que pueden cuantificarse.

**[0046]** Una CU hoja puede incluir una o más unidades de predicción (PU). En general, una PU representa una zona 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. Además, una PU incluye datos relacionados con la predicción. Por ejemplo, cuando la PU está codificada en modo intra, pueden incluirse datos para la PU en un árbol cuádruple residual (RQT), que puede incluir datos que describen un modo de predicción intra 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.

**[0047]** Una CU hoja que tiene una o más PU también puede incluir una o más unidades de transformada (TU). Las unidades de transformada pueden especificarse usando un RQT (también denominado estructura de árbol cuaternario de TU), como se ha expuesto anteriormente. Por ejemplo, un indicador de división puede indicar si una CU hoja está dividida en cuatro unidades de transformada. A continuación, cada unidad de transformada puede dividirse adicionalmente en sub-TU adicionales. Cuando una TU no se divide adicionalmente, puede denominarse una TU hoja. En general, en lo que respecta a la intracodificación, todas las TU hojas que pertenecen a una CU hoja comparten el mismo modo de predicción intra. Es decir, el mismo modo de predicción intra se aplica en general para calcular valores predichos para todas las TU de una CU hoja. En lo que respecta a la intracodificación, un codificador de vídeo puede calcular un valor residual para cada TU hoja usando el modo de predicción intra, como una diferencia entre la parte de la CU correspondiente a la TU y el bloque original. Una TU no está necesariamente limitada al tamaño de una PU. De este modo, las TU pueden ser mayores o menores que una PU. En lo que respecta a la intracodificación, una PU puede estar co-situada 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.

**[0048]** Además, las TU de las CU hojas también pueden asociarse a respectivas estructuras de datos de árbol cuádruple, denominadas árboles cuádruples residuales (RQT). Es decir, una CU hoja puede incluir un árbol cuaternario que indica cómo la CU hoja está dividida en las 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 arbolado (o LCU). Las TU del RQT que no están divididas se denominan TU hojas. En general, esta divulgación usa los términos CU y TU para hacer referencia a una CU hoja y a una TU hoja, respectivamente, a no ser que se indique lo contrario.

**[0049]** Una secuencia de vídeo incluye típicamente una serie de tramas o imágenes de vídeo. Un grupo de imágenes (GOP) comprende, en general, una serie de una o más de las imágenes de vídeo. Un GOP puede incluir datos sintácticos en una cabecera del GOP, una cabecera de una o más de las imágenes, o en otras ubicaciones, que describen una serie de imágenes incluidas en el GOP. Cada fragmento de una imagen puede incluir datos sintácticos de fragmento que describen un modo de codificación para el fragmento respectivo. El codificador de vídeo 20 actúa típicamente sobre bloques de vídeo dentro de fragmentos de vídeo individuales con el fin de codificar los datos de vídeo. Un bloque de vídeo puede corresponder a un nodo de codificación dentro de una CU. Los bloques de vídeo pueden tener tamaños fijos o variables y pueden diferir en tamaño de acuerdo a una norma de codificación especificada.

**[0050]** En un ejemplo, el HM soporta predicción en diversos tamaños de PU. Si se supone que el tamaño de una CU particular es  $2N \times 2N$ , el HM soporta predicción intra en tamaños de PU de  $2N \times 2N$  o  $N \times N$  y predicción inter en tamaños de PU simétricas de  $2N \times 2N$ ,  $2N \times N$ ,  $N \times 2N$  o  $N \times N$ . El HM también soporta la división asimétrica para predicción inter en tamaños de PU de  $2N \times nU$ ,  $2N \times nD$ ,  $nL \times 2N$  y  $nR \times 2N$ . En la división asimétrica, una dirección de una CU no está dividida, mientras que la otra dirección está dividida en un 25 % y un 75 %. La parte de la CU correspondiente a la división del 25 % está indicada por una «n» seguida por una indicación de «arriba», «abajo», «izquierda» o «derecha». Así pues, por ejemplo, « $2N \times nU$ » se refiere a una CU de tamaño  $2N \times 2N$  que está dividida horizontalmente, con una PU de tamaño  $2N \times 0,5N$  encima y una PU de tamaño  $2N \times 1,5N$  debajo.

**[0051]** En la presente divulgación, « $N \times N$ » y «N por N» pueden usarse indistintamente para hacer referencia a las dimensiones de píxeles de un bloque de vídeo en términos de dimensiones verticales y horizontales, por ejemplo,  $16 \times 16$  píxeles o 16 por 16 píxeles. En general, un bloque de  $16 \times 16$  tendrá 16 píxeles en una dirección vertical ( $y = 16$ ) y 16 píxeles en una dirección horizontal ( $x = 16$ ). Asimismo, un bloque de tamaño  $N \times N$  tiene, en general, N píxeles en

una dirección vertical y  $N$  píxeles en una dirección horizontal, donde  $N$  representa un valor entero no negativo. Los píxeles de un bloque se pueden disponer en filas y columnas. Además, no es necesario que los bloques tengan necesariamente el mismo número de píxeles en la dirección horizontal que en la dirección vertical. Por ejemplo, los bloques pueden comprender  $N \times M$  píxeles, donde  $M$  no es necesariamente igual a  $N$ .

5 **[0052]** Tras la codificación con predicción intra o predicción inter, 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 sintácticos que describen un procedimiento o modo de generación de datos de píxeles predictivos en el dominio espacial (también denominado el dominio de píxeles) y las TU pueden comprender coeficientes en el dominio de transformada, tras la aplicación de una transformada, por ejemplo, una transformada de coseno discreta (DCT), una transformada entera, una transformada de ondículas o una transformada conceptualmente similar, a los datos de vídeo residuales. Los datos residuales pueden corresponder a diferencias de píxeles entre píxeles de la imagen no codificada y valores de predicción correspondientes a las PU. El codificador de vídeo 20 puede formar las TU incluyendo los datos residuales para la CU y, a continuación, transformar las TU para generar coeficientes de transformada para la CU.

15 **[0053]** Tras cualquier transformada para generar coeficientes de transformada, el codificador de vídeo 20 puede realizar la cuantificación de los coeficientes de transformada. La cuantificación se refiere, en general, a un proceso en el que unos coeficientes de transformada se cuantifican para reducir posiblemente la cantidad de datos usados para representar los coeficientes, proporcionando compresión adicional. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos, o a la totalidad, de los coeficientes. Por ejemplo, un valor de  $n$  bits puede redondearse a la baja hasta un valor de  $m$  bits durante la cuantificación, donde  $n$  es mayor que  $m$ .

25 **[0054]** Después de la cuantificación, el codificador de vídeo puede recorrer los coeficientes de transformada, produciendo un vector unidimensional a partir de la matriz bidimensional que incluye los coeficientes de transformada cuantificados. El recorrido puede estar diseñado para colocar los coeficientes de energía más alta (y por lo tanto de menor frecuencia) al frente de la formación y para colocar los coeficientes de energía más baja (y por lo tanto de mayor frecuencia) al final de la formación. En algunos ejemplos, el codificador de vídeo 20 puede utilizar un orden de exploración predefinido para explorar los coeficientes de transformada cuantificados, para producir un vector serializado que se pueda codificar por entropía. En otros ejemplos, el codificador de vídeo 20 puede realizar una exploración adaptativa. Después de recorrer los coeficientes de transformada cuantificados para formar un vector unidimensional, el codificador de vídeo 20 puede realizar la codificación por entropía del vector unidimensional, por ejemplo, de acuerdo a la codificación de longitud variable adaptable al contexto (CAVLC), la codificación aritmética binaria adaptable al contexto (CABAC), la codificación aritmética binaria adaptable al contexto basada en la sintaxis (SBAC), la codificación por entropía por división de intervalos de probabilidad (PIPE) o con otra metodología de codificación por entropía. El codificador de vídeo 20 también puede realizar la codificación por entropía de los elementos sintácticos asociados a los datos de vídeo codificados, para su uso por el descodificador de vídeo 30 en la descodificación de los datos de vídeo.

35 **[0055]** Para realizar la codificación CABAC, el codificador de vídeo 20 puede asignar un contexto, dentro de un modelo de contexto, a un símbolo a transmitir. El contexto se puede referir, por ejemplo, a si los valores contiguos del símbolo son distintos de cero o no. Para realizar la CAVLC, el codificador de vídeo 20 puede seleccionar un código de longitud variable para un símbolo a transmitir. Las palabras de código en la VLC se pueden construir de modo que los códigos relativamente más cortos correspondan a símbolos más probables, mientras que los códigos más largos correspondan a símbolos menos probables. De esta forma, el uso de la VLC puede lograr un ahorro en bits con respecto, por ejemplo, al uso de palabras de código de igual longitud para cada símbolo que se va a transmitir. La determinación de la probabilidad se puede basar en un contexto asignado al símbolo.

45 **[0056]** En esta sección, se analizarán las técnicas de codificación de profundidad de múltiples visualizaciones y de múltiples visualizaciones plus. Inicialmente, se analizarán las técnicas de MVC. Como se ha indicado anteriormente, MVC es una ampliación de la norma ITU-T H.264/AVC. En MVC, los datos para una pluralidad de visualizaciones se codifican en orden de primer tiempo, y por consiguiente, la disposición de orden de descodificación se denomina codificación de primer tiempo. En particular, se pueden codificar componentes de visualización (es decir, imágenes) para cada una de la pluralidad de visualizaciones en una instancia de tiempo común, y a continuación se puede codificar otro conjunto de componentes de visualización para una instancia de tiempo diferente, y así sucesivamente. Una unidad de acceso puede incluir imágenes codificadas de todas las visualizaciones para una instancia de tiempo de salida. Debe comprenderse que el orden de descodificación de las unidades de acceso no es necesariamente idéntico al orden de salida (o de visualización).

55 **[0057]** La FIG. 2 muestra un típico orden de descodificación de MVC (es decir, orden del flujo de bits). La disposición de orden de descodificación se denomina codificación de tiempo primero. Se debe tener en cuenta que el orden de descodificación de las unidades de acceso puede no ser idéntico al orden de salida o de visualización. En la FIG. 2, S0 a S7 se refieren, cada uno, a diferentes visualizaciones del vídeo de visualización múltiple. Cada una entre T0 a T8 representa una instancia de tiempo de salida. Una unidad de acceso puede incluir las imágenes codificadas de todas las visualizaciones para una instancia de tiempo de salida. Por ejemplo, una primera unidad de acceso puede incluir todas las visualizaciones S0 a S7 para la instancia de tiempo T0, una segunda unidad de acceso puede incluir todas las visualizaciones S0 a S7 para la instancia de tiempo T1 y así sucesivamente.

[0058] En aras de la brevedad, la divulgación puede utilizar las siguientes definiciones:

**componente de visualización:** Una *representación codificada* de una *visualización* en una *única unidad de acceso*. Cuando una visualización incluye tanto representaciones de textura como de profundidad codificadas, un componente de visualización se compone de un componente de visualización de textura y un componente de visualización de profundidad.

**componente de visualización de textura:** Una *representación codificada* de la textura de una visualización en una *única unidad de acceso*.

**componente de visualización de profundidad:** Una *representación codificada* de la profundidad de una visualización en una *única unidad de acceso*.

[0059] En la FIG. 2, cada una de las visualizaciones incluye conjuntos de imágenes. Por ejemplo, la visualización S0 incluye un conjunto de imágenes 0, 8, 16, 24, 32, 40, 48, 56 y 64, la visualización S1 incluye el conjunto de imágenes 1, 9, 17, 25, 33, 41, 49, 57 y 65, y así sucesivamente. Cada conjunto incluye dos imágenes: una imagen se denomina componente de visualización de textura y la otra imagen se denomina componente de visualización de profundidad. El componente de visualización de textura y el componente de visualización de profundidad dentro de un conjunto de imágenes de una visualización se pueden considerar como correspondientes entre sí. Por ejemplo, el componente de visualización de textura dentro de un conjunto de imágenes de una visualización se considera correspondiente al componente de visualización de profundidad dentro del conjunto de las imágenes de la visualización, y viceversa (es decir, el componente de visualización de profundidad corresponde a su componente de visualización de textura en el conjunto, y viceversa). Como se usa en esta divulgación, un componente de visualización de textura que corresponde a un componente de visualización de profundidad se puede considerar como el componente de visualización de textura, y el componente de visualización de profundidad como parte de una misma visualización de una única unidad de acceso.

[0060] El componente de visualización de textura incluye el contenido de imagen real que se muestra. Por ejemplo, el componente de visualización de textura puede incluir los componentes de luma (Y) y croma (Cb y Cr). El componente de visualización de profundidad puede indicar profundidades relativas de los píxeles en su correspondiente componente de visualización de textura. Como un ejemplo, el componente de visualización de profundidad es una imagen en escala de grises que incluye solamente valores de luma. En otras palabras, el componente de visualización de profundidad puede que no transmita ningún contenido de imagen, sino que, en cambio, proporcione una cierta medida de las profundidades relativas de los píxeles en el componente de visualización de textura.

[0061] Por ejemplo, un píxel blanco puro en el componente de visualización de profundidad indica que su correspondiente píxel, o correspondientes píxeles, en el correspondiente componente de visualización de textura, está(n) más cerca, desde la perspectiva del observador, y un píxel negro puro en el componente de visualización de profundidad indica que su correspondiente píxel, o correspondientes píxeles, en el correspondiente componente de visualización de textura, está(n) más lejos, desde la perspectiva del observador. Los diversos tonos de gris entre negro y blanco indican diferentes niveles de profundidad. Por ejemplo, un píxel muy gris en el componente de visualización de profundidad indica que su píxel correspondiente en el componente de visualización de textura está más alejado que un píxel ligeramente gris en el componente de visualización de profundidad. Dado que solamente es necesaria la escala de grises para identificar la profundidad de los píxeles, el componente de visualización de profundidad no necesita incluir componentes de croma, ya que los valores de color para el componente de visualización de profundidad pueden no tener ningún sentido.

[0062] El componente de visualización de profundidad que utiliza solo valores de luma (por ejemplo, valores de intensidad) para identificar la profundidad se proporciona con fines ilustrativos y no debería considerarse limitativa. En otros ejemplos, se puede utilizar cualquier técnica para indicar las profundidades relativas de los píxeles en el componente de visualización de textura.

[0063] La FIG. 3 muestra una estructura de predicción MVC típica (que incluye tanto la predicción entre imágenes dentro de cada visualización como la predicción entre visualizaciones) para la codificación de vídeo de múltiples visualizaciones. Las direcciones de predicción en la FIG. 3 se indican mediante flechas, con el objeto apuntado utilizando el objeto desde el que se apunta como referencia de predicción. En la MVC, la predicción entre visualizaciones tiene soporte de la compensación de movimiento de disparidad, que usa la sintaxis de la compensación de movimiento de la H.264/AVC, pero permite que una imagen en una visualización diferente sea usada como imagen de referencia.

[0064] En el ejemplo de la FIG. 3, se ilustran seis visualizaciones (que tienen ID de visualización "S0" a "S5") y se ilustran doce ubicaciones temporales ("T0" a "T11") para cada visualización. Es decir, cada fila en la FIG. 3 corresponde a una visualización, mientras que cada columna indica una ubicación temporal.

**[0065]** Aunque la MVC tiene una denominada visualización de base que es descodificable mediante los descodificadores de la H.264/AVC y los pares de visualizaciones estéreo podrían soportarse también por MVC, la ventaja de la MVC es que podría soportar un ejemplo que usa más de dos visualizaciones como una entrada de vídeo 3D y descodifica este vídeo 3D representado por las múltiples visualizaciones. Un representador de un cliente que tiene un descodificador de MVC puede esperar contenido de vídeo 3D con múltiples visualizaciones.

**[0066]** Las imágenes en la FIG. 3 se indican en la intersección de cada fila y cada columna. La norma H.264/AVC puede usar el término trama para representar una parte del vídeo. Esta divulgación puede usar los términos imagen y trama de forma intercambiable.

**[0067]** Las imágenes en la FIG. 3 se ilustran usando un bloque que incluye una letra, designando la letra si la imagen correspondiente está intracodificada (es decir, es una imagen I), o intercodificada en una dirección (es decir, como una imagen P) o en múltiples direcciones (es decir, como una imagen B). En general, las predicciones se indican mediante flechas, donde las imágenes a las que se apunta usan la imagen desde la que se apunta como referencia de predicción. Por ejemplo, la imagen P de la visualización S2 en la ubicación temporal T0 se predice a partir de la imagen I de la visualización S0 en la ubicación temporal T0.

**[0068]** Al igual que con la codificación de vídeo de visualización única, las imágenes de una secuencia de vídeo de codificación de vídeo de múltiples visualizaciones pueden codificarse predictivamente con respecto a imágenes en diferentes ubicaciones temporales. Por ejemplo, la imagen b de la visualización S0 en la ubicación temporal T1 tiene una flecha apuntando a la misma desde la imagen I de la visualización S0 en la ubicación temporal T0, indicando que la imagen b se predice a partir de la imagen I. Adicionalmente, sin embargo, en el contexto de la codificación de vídeo de múltiples visualizaciones, las imágenes se pueden predecir entre visualizaciones. Es decir, un componente de visualización puede usar los componentes de visualización en otras visualizaciones como referencia. En la MVC, por ejemplo, la predicción entre visualizaciones se realiza como si el componente de visualización de otra visualización fuera una referencia de predicción inter. Las referencias potenciales entre visualizaciones se señalizan en la extensión de la MVC del conjunto de parámetros de secuencia (SPS) y pueden ser modificadas por el proceso de construcción de la lista de imágenes de referencia, que habilita el ordenamiento flexible de las referencias de predicción inter o de predicción entre visualizaciones. La predicción entre visualizaciones también es una característica de la ampliación propuesta de visualización múltiple de la HEVC, incluyendo la 3D-HEVC (visualización múltiple más profundidad).

**[0069]** La FIG. 3 proporciona varios ejemplos de predicción entre visualizaciones. Las imágenes de la visualización S1, en el ejemplo de la FIG. 3, se ilustran como predichas a partir de imágenes en diferentes ubicaciones temporales de la visualización S1, así como predichas entre visualizaciones a partir de imágenes de las visualizaciones S0 y S2 en las mismas ubicaciones temporales. Por ejemplo, la imagen b de la visualización S1 en la ubicación temporal T1 se predice a partir de cada una de las imágenes B de la visualización S1 en las ubicaciones temporales T0 y T2, así como las imágenes b de las visualizaciones S0 y S2 en la ubicación temporal T1.

**[0070]** En algunos ejemplos, la FIG. 3 puede verse como una ilustración de los componentes de la visualización de textura. Por ejemplo, las imágenes I, P, B y b, ilustradas en la FIG. 2, se pueden considerar componentes de visualización de textura para cada una de las visualizaciones. De acuerdo a las técnicas descritas en esta divulgación, para cada uno de los componentes de visualización de textura ilustrados en la FIG. 3 existe un correspondiente componente de visualización de profundidad. En algunos ejemplos, los componentes de visualización de profundidad se pueden predecir de una manera similar a la ilustrada en la FIG. 3 para los correspondientes componentes de visualización de textura.

**[0071]** La codificación de dos visualizaciones también podría soportarse con MVC. Una de las ventajas de la MVC es que un codificador de MVC podría tomar más de dos visualizaciones como una entrada de vídeo 3D y un descodificador de MVC puede descodificar tal representación de múltiples visualizaciones. Por lo tanto, cualquier renderizador con un descodificador MVC puede esperar contenidos de vídeo 3D con más de dos visualizaciones.

**[0072]** En la MVC, se permite la predicción entre visualizaciones entre imágenes en la misma unidad de acceso (es decir, con la misma instancia de tiempo). Cuando se codifica una imagen de una de las visualizaciones no básicas, se puede añadir una imagen a una lista de imágenes de referencia si está en una visualización diferente pero presenta una misma instancia de tiempo. Una imagen de referencia entre visualizaciones puede disponerse en cualquier posición de una lista de imágenes de referencia, tal como cualquier imagen de referencia de predicción inter. Como se muestra en la FIG. 3, un componente de visualización puede usar los componentes de visualización en otras visualizaciones como referencia. En la MVC, la predicción entre visualizaciones se realiza como si el componente de visualización en otra visualización fuera una referencia de predicción inter.

**[0073]** A continuación se describen algunas técnicas de HEVC relevantes relacionadas con la predicción inter que se pueden usar con la codificación de múltiples visualizaciones y/o la codificación de múltiples visualizaciones (MV-HEVC) con profundidad (3D-HEVC). La primera técnica para el análisis es la construcción de la lista de imágenes de referencia para la predicción inter.

**[0074]** La codificación de una PU usando predicción inter implica calcular un vector de movimiento entre un bloque

actual (por ejemplo, PU) y un bloque en una trama de referencia. Los vectores de movimiento se calculan a través de un proceso denominado estimación de movimiento (o búsqueda de movimiento). Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de una unidad de predicción en una trama actual con respecto a una muestra de referencia de una trama de referencia. Una muestra de referencia puede ser un bloque del que se descubra que se adapta estrechamente a la parte de la CU que incluye la PU codificada, en términos de diferencia de píxeles, que puede determinarse mediante la suma de una diferencia absoluta (SAD), una suma de diferencias al cuadrado (SSD) u otras métricas de diferencia. La muestra de referencia puede producirse en cualquier lugar dentro de una trama de referencia o de una trama de referencia. En algunos ejemplos, la muestra de referencia puede producirse en una posición de píxel fraccionaria. Al encontrar una parte de la trama de referencia que mejor se adapta a la parte actual, el codificador determina el vector de movimiento actual para el bloque actual como la diferencia en la ubicación del bloque actual a la parte correspondiente en la trama de referencia (por ejemplo, desde el centro del bloque actual al centro de la parte correspondiente).

**[0075]** En algunos ejemplos, un codificador puede señalar el vector de movimiento para cada bloque en el flujo de bits de vídeo codificado. El vector de movimiento señalado se usa por el descodificador para realizar una compensación de movimiento para descodificar los datos de vídeo. Sin embargo, la señalización del vector de movimiento original puede dar como resultado directamente una codificación menos eficiente, ya que se necesita típicamente una gran cantidad de bits para transmitir la información.

**[0076]** En algunos casos, en lugar de señalar directamente el vector de movimiento original, el codificador puede predecir un vector de movimiento para cada división, es decir, para cada PU. Al realizar esta predicción de vector de movimiento, el codificador puede seleccionar un conjunto de candidatos de vector de movimiento determinados a partir de bloques contiguos espacialmente en la misma trama que el bloque actual o un candidato de vector de movimiento temporal determinado a partir de un bloque que se encuentra en un mismo lugar en una trama de referencia (es decir, una trama que no sea la trama actual). El codificador de vídeo 20 puede realizar la predicción del vector de movimiento, y si es necesario, señalar un índice a una imagen de referencia para predecir el vector de movimiento, en lugar de señalar un vector de movimiento original, para reducir la velocidad de bits en la señalización. Los candidatos de vector de movimiento de los bloques contiguos espacialmente se pueden denominar candidatos MVP espaciales, mientras que los candidatos de vector de movimiento de bloques que se encuentran en el mismo lugar en otra trama de referencia se pueden denominar candidatos temporales MVP.

**[0077]** Se proponen dos modos o tipos de predicción de vector de movimiento diferentes en la norma HEVC. Un modo se denomina modo de "fusión". El otro modo se denomina predicción adaptativa de vectores de movimiento (AMVP).

**[0078]** En el modo de fusión, el codificador de vídeo 20 ordena al descodificador de vídeo 30 a través de la señalización de flujo de bits de sintaxis de predicción, copiar un vector de movimiento, un índice de referencia (identificando una trama de referencia, en una lista de imágenes de referencia dada, hacia la cual apunta el vector de movimiento) y la dirección de predicción de movimiento (que identifica la lista de imágenes de referencia (Lista 0 o Lista 1), es decir, en términos de si la trama de referencia precede o sigue temporalmente a la trama actual) de un vector de movimiento seleccionado candidato para un bloque actual de la trama. Esto se lleva a cabo señalando en el flujo de bits un índice en una lista candidata del vector de movimiento que identifica el candidato de vector de movimiento seleccionado (es decir, el candidato de MVP espacial particular o candidato de MVP temporal).

**[0079]** Por lo tanto, para el modo de fusión, la sintaxis de predicción puede incluir un indicador que identifica el modo (en este caso, el modo de "fusión") y un índice que identifica el candidato de vector de movimiento seleccionado. En algunos casos, el candidato de vector de movimiento estará en un bloque causal en referencia al bloque actual. Es decir, el candidato de vector de movimiento ya habrá sido descodificado por el descodificador de vídeo 30. Como tal, el descodificador de vídeo 30 ya ha recibido y/o determinado el vector de movimiento, el índice de referencia y la dirección de predicción de movimiento para el bloque causal. En consecuencia, el descodificador 30 de vídeo puede recuperar simplemente el vector de movimiento, el índice de referencia y la dirección de predicción de movimiento asociados con el bloque causal de la memoria y copiar estos valores como la información de movimiento para el bloque actual. Para reconstruir un bloque en modo de fusión, el descodificador 30 de vídeo obtiene el bloque predictivo usando la información de movimiento obtenida para el bloque actual, y agrega los datos residuales al bloque predictivo para reconstruir el bloque codificado. Para el modo de salto, se genera la misma lista de candidatos de fusión pero no se señala ningún residuo. Para simplificar, dado que el modo de salto tiene el mismo proceso de obtención de vectores de movimiento que el modo de fusión, todas las técnicas descritas en el presente documento se aplican tanto a los modos de fusión como de salto.

**[0080]** En AMVP, el codificador de vídeo 20 ordena al descodificador de vídeo 30, a través de señalización de flujo de bits, copiar solo el vector de movimiento del bloque candidato y usar el vector copiado como predictor para el vector de movimiento del bloque actual, y señala la diferencia del vector de movimiento (MVD). La trama de referencia y la dirección de predicción asociados con el vector de movimiento del bloque actual se señalan por separado. Un MVD es la diferencia entre el vector de movimiento actual para el bloque actual y un predictor de vector de movimiento obtenido a partir de un bloque candidato. En este caso, el codificador de vídeo 20, usando la estimación de movimiento, determina un vector de movimiento real para el bloque a codificar, y luego determina la diferencia entre el vector de movimiento real y el predictor de vector de movimiento como el valor de MVD. De esta forma, el

descodificador de vídeo 30 no usa una copia exacta del vector de movimiento candidato como el vector de movimiento actual, como en el modo de fusión, sino que puede usar un candidato de vector de movimiento que puede estar "cerca" en valor al vector de movimiento actual determinado a partir de la estimación de movimiento y agregar el MVD para reproducir el vector de movimiento actual. Para reconstruir un bloque en modo AMVP, el descodificador agrega los datos residuales correspondientes para reconstruir el bloque codificado.

**[0081]** En la mayoría de las circunstancias, el MVD requiere menos bits para señalar que todo el vector de movimiento actual. Como tal, AMVP permite una señalización más precisa del vector de movimiento actual mientras se mantiene la eficiencia de codificación antes que el envío del vector de movimiento completo. Por el contrario, el modo de fusión no permite la especificación de un MVD, y como tal, el modo de fusión sacrifica la precisión de la señalización del vector de movimiento para una mayor eficiencia de señalización (es decir, menos bits). La sintaxis de predicción para AMVP puede incluir un indicador para el modo (en este caso, el indicador AMVP), el índice para el bloque candidato, el MVD entre el vector de movimiento actual y el vector de movimiento predictivo del bloque candidato, el índice de referencia y la dirección de predicción de movimiento.

**[0082]** La FIG. 4 muestra un conjunto de ejemplo de bloques candidatos 120 que se pueden usar tanto en modo de fusión como en modo AMVP. En este ejemplo, los bloques candidatos se encuentran en las posiciones espaciales de la parte inferior izquierda (A0) 121, izquierda (A1) 122, izquierda arriba (B2) 125, arriba (B1) 124 y derecha arriba (B0) 123, y en la posición (posiciones) temporal(es) (T) 126. En este ejemplo, el bloque candidato izquierdo 122 está adyacente al borde izquierdo del bloque actual 127. El borde inferior del bloque izquierdo 122 está alineado con el borde inferior del bloque actual 127. El bloque anterior 124 está adyacente al borde superior del bloque actual 127. El borde derecho del bloque anterior 124 está alineado con el borde derecho del bloque actual 127.

**[0083]** Los bloques  $A_0$ ,  $A_1$ ,  $B_0$ ,  $B_1$  y  $B_2$  se denominan comúnmente bloques contiguos espacialmente o PU contiguos espacialmente. El codificador de vídeo 20/30 puede determinar las ubicaciones de los bloques contiguos espacialmente al determinar primero la ubicación de la muestra de luma superior izquierda de la PU actual (por ejemplo, el bloque actual 127 en la FIG. 4) con referencia a la muestra superior izquierda de la imagen actual. Esta ubicación de luma se denominará ubicación  $(xP, yP)$ . El codificador de vídeo 20/30 también puede determinar el ancho de la PU de luma (abreviado nPSW) y la altura de la PU de luma (abreviado nPSH). Basándose en la ubicación de la muestra de luma superior izquierda de la PU actual y la altura y el ancho de la PU actual, la ubicación de una muestra de luma de los bloques contiguos espaciales se puede determinar de la siguiente manera:

el bloque A0 incluye la muestra de luma en la ubicación  $(xP - 1, yP + nPSH)$ ;

el bloque A1 incluye la muestra de luma en la ubicación  $(xP - 1, yP + nPSH - 1)$ ;

el bloque B0 incluye la muestra de luma en la ubicación  $(xP + nPSW, yP - 1)$ ;

el bloque B1 incluye la muestra de luma en la ubicación  $(xP + nPSW - 1, yP - 1)$ ;

el bloque B2 incluye la muestra de luma en la ubicación  $(xP - 1, yP - 1)$ .

Los bloques contiguos espaciales pueden ser del mismo tamaño o pueden tener diferentes tamaños que la PU actual.

**[0084]** Una lista de candidatos para su uso en el modo de fusión se puede obtener usando los bloques contiguos espaciales analizados anteriormente. Los vectores de movimiento y los índices de referencia para ambas listas de referencia se pueden obtener tal como se especifica en los siguientes pasos ordenados, que son un ejemplo de un proceso de obtención para candidatos de fusión espacial. Como parte del proceso de obtención, el codificador de vídeo 20 y el descodificador de vídeo 30 comprueban la información de movimiento de las PU contiguas espaciales en el siguiente orden:  $A_1$ ,  $B_1$ ,  $B_0$ ,  $A_0$  o  $B_2$ . El codificador de vídeo 20 y el descodificador de vídeo 30 pueden realizar un recorte restringido para reducir o eliminar que los candidatos duplicados se agreguen a la lista de candidatos. El codificador de vídeo 20/30 puede obtener los candidatos espaciales que se incluirán en la lista de candidatos del modo de fusión de la siguiente manera:

- El codificador de vídeo 20/30 inserta  $A_1$  en la lista de candidatos si  $A_1$  está disponible.

- Si  $B_1$  y  $A_1$  tienen los mismos vectores de movimiento y los mismos índices de referencia, entonces el codificador de vídeo 20/30 no inserta  $B_1$  en la lista de candidatos. Si  $B_1$  y  $A_1$  no tienen los mismos vectores de movimiento y los mismos índices de referencia, entonces el codificador de vídeo 20/30 inserta  $B_1$  en la lista de candidatos, si  $B_1$  está disponible.

- Si  $B_0$  y  $B_1$  tienen los mismos vectores de movimiento y los mismos índices de referencia, entonces el codificador de vídeo 20/30 no inserta  $B_0$  en la lista de candidatos. Si  $B_0$  y  $B_1$  no tienen los mismos vectores de movimiento y los mismos índices de referencia, entonces el codificador de vídeo 20/30 inserta  $B_0$  en la lista de candidatos, si  $B_0$  está disponible.

- Si  $A_0$  y  $A_1$  tienen los mismos vectores de movimiento y los mismos índices de referencia, entonces el codificador de vídeo 20/30 no inserta  $A_0$  en la lista de candidatos. Si  $A_0$  y  $A_1$  no tienen los mismos vectores de movimiento y los mismos índices de referencia, entonces el codificador de vídeo 20/30 inserta  $A_0$  en la lista de candidatos, si  $A_0$  está disponible.

5 - El codificador de vídeo 20/30 inserta  $B_2$  en la lista de candidatos cuando no se cumplen las dos condiciones siguientes:

-  $B_2$  y  $B_1$  o  $B_2$  y  $A_1$  tienen los mismos vectores de movimiento y los mismos índices de referencia.

10 - Los cuatro candidatos de fusión espaciales de fusión obtenidos a partir de  $A_1$ ,  $B_1$ ,  $B_0$ ,  $A_0$  se incluyen en la lista de candidatos.

15 **[0085]** El codificador de vídeo 20/30 puede obtener el candidato temporal que se debe incluir en la lista de candidatos del modo de fusión de la siguiente manera:

- Para obtener el candidato de fusión temporal, el codificador de vídeo 20/30 primero identifica una imagen que se encuentra en el mismo lugar. Si la imagen actual es un fragmento B, se señala un elemento sintáctico "collocated\_from\_I0\_flag" en la cabecera del fragmento para indicar si la imagen que se encuentra en el mismo lugar es de RefPicList0 o RefPicList1.

20 - Después de que el codificador de vídeo 20/30 identifica una lista de imágenes de referencia, el codificador de vídeo 20/30 usa el elemento sintáctico collocated\_ref\_idx, señalado en la cabecera de fragmento, para identificar la imagen en la imagen en la lista.

25 - El codificador de vídeo 20/30 luego identifica una PU que se encuentra en el mismo lugar al comprobar la imagen que se encuentra en el mismo lugar. El codificador de vídeo 20/30 utiliza 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 como el candidato temporal.

30 - Cuando los vectores de movimiento identificados por el proceso anterior se utilizan para generar un candidato de movimiento para el modo de fusión, el codificador de vídeo 20/30 puede necesitar escalar los vectores de movimiento basándose en si la ubicación temporal (reflejada por el recuento de orden de la imagen, es decir, POC).

35 - En HEVC, el PPS incluye un flag enable\_temporal\_mv\_flag. Cuando una imagen particular con temporal\_id igual a 0 se refiere a un PPS que tiene enable\_temporal\_mv\_flag igual a 0, el codificador de vídeo 20/30 puede marcar todas las imágenes de referencia en el DPB "sin usar para la predicción del vector de movimiento temporal" y ningún vector de movimiento de las imágenes anteriores a esa imagen particular en orden de decodificación se usaría como un predictor de vector de movimiento temporal en la decodificación de la imagen particular o una imagen después de la imagen particular en orden de decodificación.

40 **[0086]** Si el fragmento actual es un fragmento B y el número total de candidatos obtenidos a partir de los procesos de obtención de candidatos espaciales y temporales descritos anteriormente es menor que el número máximo de candidatos y mayor que 1, entonces el codificador de vídeo 20/30 puede invocar un proceso para generar candidatos bi-predictivos. El codificador de vídeo 20/30 puede obtener los candidatos bi-predictivos que se incluirán en la lista de candidatos del modo de fusión de la siguiente manera:

45 - Basándose en la Tabla 1 siguiente, para generar un candidato bi-predictivo combinado con índice comblIdx, la información de movimiento RefList0 (MotList0) de la lista de candidatos con entrada igual a I0CandIdx si está disponible y la información de movimiento RefList1 (MotList1) de la lista de candidatos con entrada igual a I1CandIdx, si está disponible y no es idéntica a MotList0, pueden ser reutilizadas por el codificador de vídeo 20/30 como la información de movimiento RefList0 y RefList1 del candidato bi-predictivo combinado.

55 **Tabla 1 - Especificación de I0CandIdx y I1CandIdx en HEVC**

<b>comblIdx</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>
<b>I0CandIdx</b>	0	1	0	2	1	2	0	3	1	3	2	3
<b>comblIdx</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>
<b>I1CandIdx</b>	1	0	2	0	2	1	3	0	3	1	3	2

60 **[0087]** El codificador de vídeo 20/30 puede obtener los candidatos de vector de movimiento cero que se incluirán en la lista de candidatos de modo de fusión si el número total de candidatos obtenidos a partir de los tres procesos anteriores descritos anteriormente para determinar candidatos espaciales, temporales y bi-predictivos produce menos que el número máximo de candidatos. En tal caso, el codificador de vídeo 20/30 puede insertar unos vectores de

movimiento cero para cada imagen de referencia en la lista de candidatos. En HEVC WD 10, el número total de candidatos en la lista de fusión (MRG) es de hasta 5 y `five_minus_max_num_merge_cand` se indica para especificar el número máximo de candidatos de MRG restados de 5 en la cabecera de fragmento.

5 **[0088]** Para mejorar aún más la eficiencia de codificación para codificar vídeo 3D, se han introducido dos nuevas tecnologías, a saber, "predicción de movimiento entre visualizaciones" y "predicción residual entre visualizaciones" en 3D-HEVC. Para habilitar estas dos herramientas de codificación, el codificador de vídeo 20/30 está configurado para obtener vectores de disparidad para un bloque actual.

10 **[0089]** El codificador de vídeo 20/30 se puede configurar para obtener un vector de disparidad, usando una técnica denominada vector de disparidad basado en bloques contiguos (NBDV), que utiliza vectores de movimiento de disparidad a partir de bloques contiguos espaciales y temporales o un vector de disparidad obtenido a partir de bloques contiguos para determinar un vector de disparidad para un bloque actual. En NBDV, el codificador de vídeo 20/30 comprueba los vectores de movimiento de bloques contiguos espaciales o temporales en una orden de comprobación  
15 fija, y una vez que se identifica un vector de movimiento de disparidad, es decir, el vector de movimiento apunta a una imagen de referencia entre visualizaciones, el codificador de vídeo 20/30 finaliza el proceso de comprobación. El codificador de vídeo 20/30 devuelve el vector de movimiento de disparidad identificado y lo convierte en el vector de disparidad para el bloque actual que se puede usar en la predicción de movimiento entre visualizaciones y la predicción residual entre visualizaciones. Si no se encuentra un vector de movimiento de disparidad después de comprobar todos  
20 los bloques contiguos predefinidos, el codificador de vídeo 20/30 puede usar un vector de disparidad cero para la predicción de movimiento entre visualizaciones mientras que la predicción residual entre visualizaciones puede deshabilitarse para la unidad de predicción correspondiente (PU).

25 **[0090]** Los bloques utilizados por el codificador de vídeo 20/30 para NBDV incluyen bloques contiguos espaciales y bloques contiguos temporales. El codificador de vídeo 20/30 analiza hasta cinco bloques contiguos espaciales como parte de la obtención del vector de disparidad. Los cinco bloques contiguos en el espacio incluyen los bloques inferior izquierdo, izquierdo, superior derecho, superior y superior izquierdo de una PU actual, designados por  $A_0$ ,  $A_1$ ,  $B_0$ ,  $B_1$  y  $B_2$ . A este respecto, los bloques contiguos espaciales analizados como parte de NBDV pueden corresponder a los mismos bloques contiguos espaciales analizados anteriormente con respecto al modo de fusión en la FIG. 4, aunque  
30 los bloques espaciales contiguos utilizados para NBDV y la construcción de la lista de candidatos para los modos de AMVP y fusión no necesariamente tienen que ser los mismos.

**[0091]** El codificador de vídeo 20/30 también determina bloques contiguos temporales. El codificador de vídeo 20/30 puede tratar todas las imágenes de referencia de la visualización actual como imágenes candidatas, o en algunas implementaciones, el número de imágenes candidatas puede restringirse aún más, por ejemplo, a cuatro imágenes candidatas. El codificador de vídeo 20/30 primero comprueba una imagen de referencia que se encuentra en el mismo lugar y comprueba el resto de las imágenes candidatas, por ejemplo, en el orden ascendente del índice de referencia (`refIdx`). Cuando tanto `RefPicList0[refIdx]` como `RefPicList1[refIdx]` están disponibles, `RefPicListX[refIdx]` precede a la otra imagen, en el que X es igual a `collocated_from_I0_flag`.  
35

40 **[0092]** Para cada imagen candidata, el codificador de vídeo 20/30 determina tres regiones candidatas para obtener los bloques contiguos temporales. Cuando una región cubre más de un bloque de 16x16, el codificador de vídeo 20/30 puede comprobar todos los bloques de 16x16 en dicha región, por ejemplo, en el orden de escaneo de trama. Las tres regiones candidatas se definen de la forma siguiente: (1) CPU- La región que se encuentra en el mismo lugar que la PU actual o CU actual, (2) CLCU - La unidad de codificación más grande (LCU) que cubre la región que se encuentra en el mismo lugar que la PU actual, y (3) BR - Bloque 4x4 inferior derecho de CPU.  
45

**[0093]** Debe entenderse que las técnicas descritas anteriormente para comprobar bloques contiguos temporales de imágenes de referencia son un ejemplo. Puede haber otros ejemplos de imágenes de referencia que estén comprobadas.  
50

**[0094]** La FIG. 5 muestra un ejemplo de bloques que se comprueban en una imagen candidata para determinar un vector de disparidad para un bloque actual. Por ejemplo, el codificador de vídeo 20/30 puede considerar hasta dos imágenes candidatas de la visualización actual. La primera imagen candidata de la visualización actual puede ser la imagen que se encuentra en el mismo lugar identificada como parte de la construcción de la lista de candidatos (por ejemplo, el candidato temporal). La segunda imagen candidata puede ser una imagen de acceso aleatorio o la imagen de referencia con la diferencia POC más pequeña y el ID temporal más pequeño. En algunos ejemplos, el codificador de vídeo 20/30 puede comprobar primero la imagen de acceso aleatorio, y si no se encuentra ningún vector de movimiento de disparidad, comprobar la imagen de ubicación conjunta.  
55

60 **[0095]** Para cada una de estas imágenes candidatas, el codificador de vídeo 20/30 puede comprobar dos bloques. El primer bloque puede ser el bloque central (CR), que puede ser el bloque central 4x4 de la región que se encuentra en el mismo lugar que la PU actual, como se ilustra mediante la Pos. A en la FIG. 5. El segundo bloque puede ser el bloque inferior derecho (BR), que puede ser el bloque 4x4 inferior derecho de la región que se encuentra en el mismo lugar que la PU actual, como se ilustra mediante la Pos. B en la FIG. 5.  
65

**[0096]** El codificador de vídeo 20/30 se puede configurar para comprobar los bloques en un orden específico, es decir, una orden de comprobación. El codificador de vídeo 20/30 puede primero comprobar los bloques contiguos espaciales, seguidos de bloques contiguos temporales. El orden de comprobación de los cinco bloques contiguos espaciales puede, por ejemplo, definirse como A<sub>1</sub>, B<sub>1</sub>, B<sub>0</sub>, A<sub>0</sub> y B<sub>2</sub>, o cualquier otro orden similar. Para cada imagen candidata, el codificador de vídeo 20/30 puede comprobar las tres regiones candidatas en la imagen candidata en una orden de comprobación. El orden de comprobación de las tres regiones puede, por ejemplo, definirse como CPU, CLCU y BR para la primera visualización no básica, BR, CPU y CLU para la segunda visualización no básica, o cualquier otro orden similar. En otro ejemplo, para cada imagen candidata, (por ejemplo, donde hay hasta dos imágenes candidatas), los dos bloques se comprueban en orden, CR y BR para la primera visualización no básica o BR, CR para la segunda visualización no básica.

**[0097]** Como se describió anteriormente, el codificador de vídeo 20/30 puede determinar el vector de disparidad para el bloque actual al comprobar los vectores de movimiento de disparidad para los bloques contiguos, y si se encuentra un vector de movimiento de disparidad para un bloque contiguo, convirtiendo ese vector de movimiento de disparidad como el vector de disparidad del bloque actual. Además, en algunos ejemplos, se puede usar un vector de disparidad obtenido para un bloque contiguo para determinar el vector de disparidad para el bloque actual. Por ejemplo, un bloque contiguo no puede tener una predicción inter con un vector de movimiento de disparidad, pero se puede haber obtenido un vector de disparidad para el bloque contiguo durante la predicción del bloque contiguo. Este vector de disparidad obtenido se puede denominar vector de disparidad implícito (IDV) y se puede usar para determinar el vector de disparidad para el bloque actual, como cuando no se realiza predicción inter de ninguno de los bloques contiguos que se comprueban con un vector de movimiento de disparidad.

**[0098]** Por ejemplo, se genera un vector de disparidad implícita (IDV) cuando una PU emplea predicción de vector de movimiento entre visualizaciones, es decir, el candidato para los modos de AMVP o de fusión se obtiene de un bloque correspondiente en la otra visualización con la ayuda de un vector de disparidad. Dicho vector de disparidad se llama IDV. El IDV se almacena en la PU con el propósito de la obtención del vector de disparidad.

**[0099]** Como se ha descrito con mayor detalle, el codificador de vídeo 20/30 se puede configurar para realizar una obtención de candidatos entre visualizaciones. Basándose en el vector de disparidad (DV), el codificador de vídeo 20/30 puede determinar un nuevo candidato de vector de movimiento, denominado candidato entre visualizaciones, y añadir el nuevo candidato de vector de movimiento a una lista de candidatos para usar en modos AMVP y de salto/fusión.

**[0100]** Por ejemplo, un ejemplo de la obtención de candidato entre visualizaciones es un candidato de vector de movimiento predicho entre visualizaciones (IPMVC), y otro ejemplo del candidato de vector de movimiento de disparidad entre visualizaciones (IDMVC). El IPMVC incluye información de movimiento (por ejemplo, vector de movimiento y uno o más índices de referencia para listas de imágenes de referencia) para un bloque de referencia en una visualización de referencia a la que se hace referencia mediante un vector de disparidad del bloque actual, donde se determina el vector de disparidad para el bloque actual basándose en la técnica de ejemplo anterior. El IDMVC es el vector de disparidad convertido en un vector de movimiento de disparidad.

**[0101]** En algunos ejemplos, el codificador de vídeo 20/30 puede desplazar el vector de disparidad (por ejemplo, sumando o restando un valor de desviación del vector de disparidad). El codificador de vídeo 20/30 puede obtener IPMVC adicionales y los IDMVC basándose en el vector de disparidad desplazado utilizando una técnica similar a la de aquellos sin desplazar el vector de disparidad. Por ejemplo, el codificador de vídeo 20/30 puede desplazar el vector de disparidad horizontalmente en una dirección (por ejemplo, sumando o restando un valor de desviación fija del componente x del vector de disparidad).

**[0102]** El codificador de vídeo 20/30 puede identificar un bloque de referencia en una visualización de referencia identificada por el vector de disparidad desplazado el valor de desviación. La información de movimiento para el bloque de referencia identificado por el vector de disparidad desplazado el valor de desviación puede ser un ejemplo de un IPMVC adicional (es decir, además del IPMVC obtenido sin desplazar el vector de disparidad). Como otro ejemplo, el codificador de vídeo 20/30 puede desplazar el vector de disparidad y puede convertir el vector de disparidad desplazada como un IDMVC adicional (es decir, además del IDMVC obtenido sin desplazar el vector de disparidad).

**[0103]** En algunos ejemplos, la cantidad en que el codificador de vídeo 20/30 puede desplazar el vector de disparidad para obtener el IPMVC puede ser diferente de la cantidad en que el codificador de vídeo 20/30 puede desplazar el vector de disparidad para obtener el IDMVC. En otras palabras, el codificador de vídeo 20/30 puede desplazar el vector de disparidad, para el bloque actual, por un primer valor de desviación para obtener el IPMVC adicional, y desplazar el vector de disparidad, para el bloque actual, por un segundo valor de desviación diferente para obtener el IDMVC adicional.

**[0104]** El primer valor de desviación o el segundo valor de desviación pueden considerarse como un vector que incluye un valor desplazado horizontal y un valor desplazado vertical. Por ejemplo, para desplazar el vector de disparidad el primer valor de desviación para obtener el IPMVC, el codificador de vídeo 20/30 puede sumar o restar el vector de disparidad, para el bloque actual, a un primer vector que incluye el valor desplazado horizontal y el desplazamiento

vertical valor. De manera similar, para desplazar el vector de disparidad el segundo valor de desviación para obtener el IDMVC, el codificador de vídeo 20/30 puede sumar o restar el vector de disparidad, para el bloque actual, a un segundo vector que incluye el valor desplazado horizontal y el valor desplazado vertical. En las técnicas descritas en esta divulgación, el primer vector utilizado para desplazar el vector de disparidad un primer valor de desviación y el segundo vector utilizado para desplazar el vector de disparidad un segundo valor de desviación pueden ser vectores diferentes. En algunos ejemplos, el valor desplazado vertical para el primer o el segundo valor de desviación puede ser igual a 0 para lograr solo un desplazamiento horizontal en el vector de disparidad para obtener el IPMVC y el IDMVC.

**[0105]** La FIG. 6 muestra un ejemplo del proceso de obtención del candidato de vector de movimiento predicho entre visualizaciones (IPMVC). Con un vector de disparidad (por ejemplo, determinado mediante las técnicas descritas anteriormente), el codificador de vídeo 20/30 puede ubicar el bloque 142 correspondiente en una visualización diferente (por ejemplo, la visualización 0 o V0, que también es una visualización de referencia) a la PU actual 140 en la visualización codificada actualmente (visualización 1 o V1). Si el bloque correspondiente 142, también conocido como bloque de referencia, no está intracodificado ni tiene predicción entre visualizaciones, y su imagen de referencia tiene un valor POC que se encuentra en la lista de imágenes de referencia de la PU 140 actual (por ejemplo, Ref0, Lista 0; Ref0, Lista 1; Ref1, Lista 1, como se muestra en la FIG. 6), a continuación la información de movimiento (por ejemplo, al menos uno de los índices de vectores de movimiento e imágenes de referencia) para el bloque correspondiente 142 se utiliza como un vector de movimiento predicho entre visualizaciones.

**[0106]** Puede haber ciertos requisitos sobre si la información de movimiento para el bloque 142 correspondiente (por ejemplo, el bloque de referencia) puede considerarse como un IPMVC. En otras palabras, puede haber ciertos requisitos para determinar si un IPMVC está disponible. Un requisito puede ser que el vector de movimiento para el bloque correspondiente 142 sea un vector de movimiento temporal. Un vector de movimiento temporal es un vector de movimiento que se refiere a una imagen en la misma visualización. Por ejemplo, en la FIG. 6, el bloque 142 correspondiente (por ejemplo, el bloque 142 de referencia) está a la visualización 0, y para que el bloque 142 correspondiente tenga un vector de movimiento temporal, el vector de movimiento para el bloque 142 correspondiente tendría que referirse a las imágenes en la misma visualización que el bloque 142 correspondiente (por ejemplo, visualización 0). Como se ilustra en la FIG. 6, los vectores de movimiento para el bloque 142 correspondiente son vectores de movimiento temporal ya que se refieren a imágenes en la misma visualización de referencia (por ejemplo, visualización 0) como bloque 142 correspondiente.

**[0107]** Otro requisito puede ser que el vector de movimiento temporal del bloque de referencia (por ejemplo, el bloque 142 correspondiente) se refiera a una imagen en la visualización de referencia con un valor de cuenta de orden de imagen (POC) que es igual a un valor POC de una imagen en una lista de imágenes de referencia (por ejemplo, RefPicList0 o RefPicList1) para el bloque actual (por ejemplo, PU 140 actual). Como un ejemplo ilustrativo para ayudar a comprender, las listas de imágenes de referencia para el bloque 142 correspondiente pueden identificar las dos imágenes utilizadas para realizar predicción inter del bloque 142 correspondiente que se encuentra en un primer índice de referencia en RefPicList0 y un segundo índice de referencia en RefPicList1. Los valores de POC para las dos imágenes que se usan para predecir el bloque 142 correspondiente pueden ser POC X y POC Y. Para que se considere que el IPMVC está disponible, al menos una lista de imágenes de referencia (es decir, RefPicList0 o RefPicList1) debe incluir imágenes con valores de POC iguales a POC X y POC Y. En otras palabras, para que se considere que el IPMVC está disponible, el vector de movimiento temporal para el bloque de referencia (por ejemplo, el bloque 142 correspondiente) debe referirse a una imagen en la visualización de referencia (por ejemplo, visualización 0) con un valor POC que es igual a un valor POC de una imagen en una lista de imágenes de referencia para el bloque actual (por ejemplo, PU actual 140).

**[0108]** Para la PU 140 actual, las imágenes con valores POC X e Y pueden no estar ubicadas en el primer índice de referencia en RefPicList0 y el segundo índice de referencia en RefPicList1, sino que pueden ubicarse en un tercer índice de referencia en RefPicList0 y un cuarto índice de referencia en RefPicList1, respectivamente. En este ejemplo, cuando la información de movimiento del bloque 142 correspondiente se agrega como candidato de vector de movimiento predicho entre visualizaciones, el codificador de vídeo 20/30 puede agregar el tercer índice de referencia en RefPicList0 y el cuarto índice de referencia en RefPicList1 como los índices de referencia.

**[0109]** Además, el vector de vídeo 20/30 puede convertir el vector de disparidad en un vector de movimiento de disparidad entre visualizaciones, que puede agregarse a la lista de candidatos de fusión en una posición diferente al IPMVC, o agregarse a la lista de candidatos de AMVP en la misma posición que el IPMVC cuando el vector de disparidad está disponible. En este contexto, se puede hacer referencia a IPMVC y al candidato de vector de movimiento de disparidad entre visualizaciones (IDMVC) como un "candidato entre visualizaciones".

**[0110]** En el modo AMVP, si el índice de imagen de referencia se refiere a una imagen de referencia temporal (por ejemplo, el índice de referencia objetivo corresponde a un vector de movimiento temporal), el codificador de vídeo 20/30 puede obtener el IPMVC y no puede obtener el IDMVC. Por ejemplo, el codificador de vídeo 20/30 puede encontrar un IPMVC al comprobar el vector de movimiento en el bloque correspondiente (es decir, el bloque 142) de la PU actual (es decir, el bloque 140) ubicado mediante el vector de disparidad. Si no está disponible (es decir, sin vector de movimiento), el codificador de vídeo 20/30 puede agregar un candidato de vector de movimiento cero a la

lista de candidatos. En el modo AMVP, si el índice de imagen de referencia se refiere a una imagen de referencia entre visualizaciones (por ejemplo, el índice de referencia objetivo corresponde a un vector de movimiento de disparidad), el codificador de vídeo 20/30 puede no obtener un IPMVC, pero en cambio, puede convertir el vector de disparidad en un vector de movimiento de disparidad entre visualizaciones y agregar el vector de disparidad convertido a la lista de candidatos (es decir, agregar el vector de movimiento de disparidad entre visualizaciones a la lista de candidatos como el IDMVC).

**[0111]** En otras palabras, si el bloque actual tiene predicción inter en modo AMVP, el codificador de vídeo 20/30 puede determinar si obtener el IPMVC o el IDMVC basándose en si el al menos un índice de imagen de referencia se refiere a una imagen de referencia temporal o una imagen de referencia entre visualizaciones. Por ejemplo, en el modo AMVP, el codificador de vídeo 20 señala un índice de imagen de referencia que el descodificador de vídeo 30 usa para identificar la imagen de referencia que se usará para la predicción inter. De esta manera, el codificador de vídeo 20/30 puede determinar si un índice de imagen de referencia se refiere a una imagen de referencia temporal o una imagen de referencia entre visualizaciones. Una imagen de referencia temporal es una imagen de referencia en la misma visualización, y una imagen de referencia entre visualizaciones es una imagen de referencia en otra visualización diferente. En algún ejemplo, si el codificador de vídeo 20/30 determina que el índice de imagen de referencia se refiere a una imagen de referencia temporal, el codificador de vídeo 20/30 puede obtener un IPMVC, y si el codificador de vídeo 20/30 determina que el índice de imagen de referencia se refiere a una imagen de referencia entre visualizaciones, el codificador de vídeo 20/30 puede obtener un IDMVC.

**[0112]** En el modo de fusión/salto, el codificador de vídeo 20/30 puede siempre insertar un IPMVC, si está disponible, antes de todos los candidatos de fusión espacial y temporal en la lista de candidatos de fusión. El codificador de vídeo 20/30 puede, por ejemplo, insertar el IPMVC antes del candidato de fusión espacial obtenido a partir de  $A_0$ , como se muestra arriba en la FIG. 4. En alguna implementación del software 3D-HTM, el vector de movimiento predicho entre visualizaciones o el vector de movimiento de disparidad entre visualizaciones siempre se inserta antes que todos los candidatos espaciales y temporales en la lista de candidatos AMVP.

**[0113]** Como se introdujo anteriormente, el codificador de vídeo 20/30 puede construir una lista de candidatos de fusión en 3D-HEVC. El codificador de vídeo 20/30 primero obtiene un vector de disparidad utilizando, por ejemplo, las técnicas NBDV descritas anteriormente. Si no se encuentra dicho vector de disparidad, el codificador de vídeo 20/30 puede utilizar un vector de disparidad cero. Con el vector de disparidad, el codificador de vídeo 20/30 construye la lista de candidatos de fusión de la siguiente manera. El codificador de vídeo 20/30 obtiene el IPMVC como se describe anteriormente y, si está disponible, inserta el IPMVC en la lista de fusión.

**[0114]** El codificador de vídeo 20/30 puede configurarse para implementar un proceso de obtención para los candidatos de fusión espacial e inserción de IDMVC. El codificador de vídeo 20/30 puede, por ejemplo, comprobar la información de movimiento de las PU contiguas espaciales en el siguiente orden:  $A_1$ ,  $B_1$ ,  $B_0$ ,  $A_0$  o  $B_2$ . El codificador de vídeo 20/30 puede realizar una purga restringida mediante los siguientes procedimientos:

- Inserte  $A_1$  en la lista de candidatos si está disponible y es diferente de IPMVC.
- Si  $B_1$  y  $A_1$ /IPMVC tienen los mismos vectores de movimiento y los mismos índices de referencia,  $B_1$  no se inserta en la lista de candidatos. De lo contrario, inserte  $B_1$  en la lista de candidatos si está disponible.
- Si  $B_0$  y  $B_1$  tienen los mismos vectores de movimiento y los mismos índices de referencia,  $B_0$  no se inserta en la lista de candidatos. De lo contrario, inserte  $B_0$  en la lista de candidatos si está disponible.
- El IDMVC se obtiene mediante el procedimiento descrito anteriormente. Si está disponible y es diferente a los candidatos obtenidos a partir de  $A_1$  y  $B_1$ , se inserta en la lista de candidatos.
- Si  $A_0$  y  $A_1$  tienen los mismos vectores de movimiento y los mismos índices de referencia,  $A_0$  no se inserta en la lista de candidatos. De lo contrario, inserte  $A_0$  en la lista de candidatos, si está disponible.
- $B_2$  se inserta en la lista de candidatos cuando no se cumple ninguna de las dos condiciones siguientes:
  - $B_2$  y  $B_1$  o  $B_2$  y  $A_1$  tienen los mismos vectores de movimiento y los mismos índices de referencia.
  - Los cuatro candidatos de fusión espacial obtenidos a partir de  $A_1$ ,  $B_1$ ,  $B_0$ ,  $A_0$  e IPMVC se incluyen en la lista de candidatos.

**[0115]** El codificador de vídeo 20/30 también se puede configurar para implementar un proceso de obtención para un candidato de fusión temporal. Este proceso de obtención puede, por ejemplo, ser el mismo procedimiento que se define en HEVC, que se analiza brevemente más arriba.

**[0116]** El codificador de vídeo 20/30 también se puede configurar para implementar un proceso de obtención para candidatos de fusión bi-predictiva combinados. Por ejemplo, si el número total de candidatos obtenidos a partir de las

dos etapas anteriores es menor que el número máximo de candidatos, el codificador de vídeo 20/30 puede realizar el mismo proceso según lo definido en la HEVC, excepto la especificación de l0CandIdx y l1CandIdx. La relación entre comblIdx, l0CandIdx y l1CandIdx se define en la siguiente tabla:

5 **Tabla 2 - Especificación de l0CandIdx y l1CandIdx en la 3D-HEVC**

comblIdx	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
l0CandIdx	0	1	0	2	1	2	0	3	1	3	2	3	0	4	1	4	2	4	3	4
l1CandIdx	1	0	2	0	2	1	3	0	3	1	3	2	4	0	4	1	4	2	4	3

10 **[0117]** El codificador de vídeo 20/30 también se puede configurar para implementar un proceso de obtención para candidatos de fusión de vectores de movimiento cero. El codificador de vídeo 20/30 puede, por ejemplo, implementar el mismo procedimiento definido en HEVC.

15 **[0118]** En el software más actual, el número total de candidatos en la lista de fusión (MRG) es de hasta 6 y se señala el elemento sintáctico "five\_minus\_max\_num\_merge\_cand" para especificar el número máximo de candidatos de MRG restados a 6 en la cabecera de fragmento. Se debe notar que five\_minus\_max\_num\_merge\_cand está en el rango de 0 a 5, inclusive. Como se indicó anteriormente, en WD10 el número total de candidatos en la lista de MRG es de hasta 5.

20 **[0119]** Además de los procedimientos descritos anteriormente, de acuerdo con otras propuestas que fusionan candidatos obtenidos a partir de un vector de disparidad, la siguiente técnica también fue propuesta en JCT3V-B0048 por Zhang et. Al, titulado "3D-CE5,h: Merge candidates derivation from disparity vector [3D-CE5,h: obtención de candidatos de fusión a partir del vector de disparidad]", 15 de octubre de 2012, disponible a partir del 4 de diciembre de 2013, [http://phenix.itsudparis.eu/jct2/doc\\_end\\_user/current\\_document.php?id=237](http://phenix.itsudparis.eu/jct2/doc_end_user/current_document.php?id=237). El bloque correspondiente (es decir, el bloque de referencia) identificado por el vector de disparidad en una visualización de referencia se utiliza para obtener IPMVC como procedimiento actual. Sus PUs izquierda y derecha de la PU que contienen el bloque correspondiente están ubicadas y se obtienen dos vectores de disparidad desplazados, si corresponde. Vector de disparidad a la izquierda (LDV): reste al vector de disparidad el ancho de la PU izquierda para el componente horizontal. Vector de disparidad desplazado a la derecha (RDV): agregue al vector de disparidad el ancho de la PU que contiene el bloque correspondiente en el componente horizontal. El uso del LDV/RDV para obtener IPMVC es el mismo que el vector de disparidad en el 3D-HEVC actual. Si el IPMVC obtenido está disponible, se agrega a la lista de candidatos de fusión; de lo contrario, el vector de disparidad desplazado se convierte en un vector de movimiento de disparidad para agregarse a la lista de candidatos de fusión.

35 **[0120]** El diseño actual de las tecnologías relacionadas con el movimiento en la codificación de múltiples visualizaciones/3DV basada en HEVC tiene varios posibles problemas debido al hecho de que el vector de disparidad obtenido podría no ser lo suficientemente preciso. Como ejemplo, un vector de disparidad impreciso puede llevar a candidatos de vector de movimiento predichos entre visualización y candidatos de vector de movimiento de disparidad entre visualizaciones imprecisos. Como otro ejemplo, la utilización de las PU izquierda/derecha del bloque correspondiente en una visualización de referencia para obtener IPMVC puede no ser fácil para la implementación del hardware ya que el acceso a las PU izquierda/derecha necesita cambiar desde el bloque correspondiente con su ancho que es variable. Como otro ejemplo, para eliminar a los candidatos reduplicados, todos los índices de referencia y vectores de movimiento deben ser comprobados, lo cual introduce demasiada complejidad.

45 **[0121]** Esta divulgación introduce varias técnicas que pueden mejorar la precisión del vector de disparidad así como la precisión de los vectores de movimiento de predicción entre visualizaciones y los vectores de movimiento de disparidad entre visualizaciones. De acuerdo con una técnica de esta divulgación, el codificador de vídeo 20/30 puede obtener más IPMVC de la visualización de referencia de vectores de disparidad desplazados con un valor de desviación fijo para todas las PU dentro de una trama/imagen/secuencia. En un ejemplo, el codificador de vídeo 20/30 puede desplazar horizontalmente el vector de disparidad en  $-4 * M$  (M es un entero positivo, como 1, 2, 4, 8) y/o  $4 * M$  (correspondiente a M píxeles). Con el vector de disparidad desplazado, el codificador de vídeo 20/30 puede usar las PU ubicadas en el lado izquierdo y/o derecho de la PU que contiene el bloque correspondiente para generar IPMVC similares al proceso de generación de IPMVC del bloque correspondiente.

55 **[0122]** De acuerdo con otra técnica de esta divulgación, el codificador de vídeo 20/30 puede convertir más IDMVC con vectores de disparidad desplazados. En un ejemplo, el valor desplazado (es decir, el valor de desviación) es  $-4 * N$  (N es un entero positivo, como 1, 2, 4, 8) y/o  $4 * N$  (correspondiente a N píxeles) horizontalmente. En un ejemplo, además, el valor desplazado (es decir, el valor de desviación) para obtener IPMVC e IDMVC podría ser diferente, es decir, N es diferente de M.

60 **[0123]** De acuerdo con otra técnica de esta divulgación, para cada mutación basada en un vector de disparidad (es decir, cada desplazamiento del vector de disparidad), el codificador de vídeo 20/30 puede generar solo un candidato adicional, y el candidato es de IPMVC basado en un primer vector de disparidad desplazada o el IDMVC basado en

un segundo vector de disparidad desplazado. Si IPMVC está disponible, el codificador de vídeo 20/30 puede usarlo como candidato adicional. Si se agrega IPMVC a la lista de candidatos de fusión, el codificador de vídeo 20/30 puede no convertir IDMVC. En un ejemplo, si IDMVC está disponible, el codificador de vídeo 20/30 puede no obtener IPMVC. En otro ejemplo, dos de ellos, si están disponibles, pueden obtenerse como candidatos adicionales.

**[0124]** En otras palabras, si el bloque actual tiene predicción inter en modo de fusión o modo de salto, el codificador de vídeo 20/30 puede determinar la disponibilidad del IPMVC, donde el IPMVC es información de movimiento (por ejemplo, vector de movimiento y uno o más índices de referencia) para un bloque de referencia en una visualización de referencia a la que se hace referencia mediante un vector de disparidad, para el bloque actual, desplazado un valor de desviación. Por ejemplo, el codificador de vídeo 20/30 puede determinar que el IPMVC está disponible si el vector de movimiento para el bloque de referencia es un vector de movimiento temporal que se refiere a una imagen en la visualización de referencia con un valor de cuenta de orden de imagen (POC) que es igual a un valor POC de una imagen en una lista de imágenes de referencia para el bloque actual.

**[0125]** En tal ejemplo, el codificador de vídeo 20/30 puede obtener el IPMVC para su posible inclusión en la lista de candidatos (sujeto a la purga que se describe a continuación), y puede evitar la obtención del IDMVC (es decir, evitar la obtención del IDMVC). Por ejemplo, en este ejemplo, el codificador de vídeo 20/30 puede priorizar IPMVC sobre IDMVC, si el bloque actual tiene predicción inter en modo de fusión o salto, donde si el IPMVC está disponible, el codificador de vídeo 20/30 puede obtener el IPMVC, y no puede obtener el IDMVC (es decir, evitar la obtención del IDMVC).

**[0126]** De acuerdo con otra técnica de esta divulgación, para cada mutación basada en un vector de disparidad, el primer vector de disparidad desplazado utilizado para la obtención de IPMVC y el segundo vector de disparidad desplazado utilizado para la conversión de IPMVC pueden tener diferentes valores de desplazamiento, donde los valores de desplazamiento indican el cantidad en la que el codificador de vídeo 20/30 cambia el vector de disparidad. Por ejemplo, el valor de desviación para el primer vector de disparidad desplazado es 16 y el desplazamiento para el segundo vector de disparidad desplazado es 4. En algunos ejemplos, ambos vectores de disparidad desplazados deben estar en dirección horizontal y en la misma dirección, lo cual significa que los valores de signo de las desviaciones son los mismos. En un ejemplo, solo hay dos mutaciones, una a la izquierda y otra a la derecha. De forma alternativa, solo hay una mutación a la izquierda. De forma alternativa, solo hay una mutación a la derecha. De acuerdo con otra técnica de esta divulgación, cada uno de los candidatos puede insertarse en cualquier posición de la lista de candidatos.

**[0127]** Como ejemplo, si el codificador de vídeo 20/30 obtiene el IPMVC, la información de movimiento para el IPMVC puede incluir un vector de movimiento (por ejemplo, un vector de movimiento temporal) para un bloque de referencia en una visualización de referencia a la que se hace referencia mediante un vector de disparidad, para el bloque actual, desplazado un primer valor de desviación. Además, el vector de movimiento del IPMVC puede referirse a una imagen en la misma visualización (por ejemplo, una imagen de referencia temporal) cuyo valor POC es igual a un valor POC de una imagen identificada en una lista de imágenes de referencia del bloque actual. Si el codificador de vídeo 20/30 obtiene el IDMVC, el IDMVC puede ser el vector de disparidad, para el bloque actual, desplazado un segundo valor de desviación diferente. En algunos ejemplos, el codificador de vídeo 20/30 puede agregar el primer valor de desviación o restar el primer valor de desviación del vector de disparidad para determinar el vector de disparidad desplazado y el primer valor de desviación. En algún ejemplo, el codificador de vídeo 20/30 puede agregar el segundo valor de desviación o restar el segundo valor de desviación del vector de disparidad para determinar el vector de disparidad desplazado el segundo valor de desviación.

**[0128]** De acuerdo con otra técnica de esta divulgación, se puede aplicar una purga basada en el tipo para cada uno de los candidatos de vector de movimiento recién agregados, incluso si se incluye el obtenido a partir del primer vector de disparidad. El codificador de vídeo 20/30 puede aplicar el proceso de purga para dos candidatos solo cuando los tipos de imágenes de referencia (imágenes de referencia entre visualizaciones o imágenes de referencia temporales) son iguales. En este caso, cuando el tipo de imagen de referencia correspondiente a RefPicList0 o RefPicList1 es diferente al de un candidato a ser comparado, no se necesita una comparación de los índices de referencia y los vectores de movimiento (es decir, el codificador de vídeo 20/30 puede no realizar la comparación de índices de referencia y vectores de movimiento si el tipo de imagen de referencia al que hace referencia un primer candidato es diferente al de un segundo candidato). El codificador de vídeo 20/30 puede aplicar el proceso de purga para dos candidatos de entrevista solo cuando ambos sean IDMVC o IPMVC.

**[0129]** Por ejemplo, además de obtener un IPMVC y un IDMVC basado en vectores de disparidad desplazados, el codificador de vídeo 20/30 puede obtener un IPMVC y un IDMVC basado en el vector de disparidad para el bloque actual sin desplazar el vector de disparidad. Como ejemplo, supongamos que un primer IPMVC es un vector de movimiento para un primer bloque de referencia al que hace referencia un vector de disparidad, para el bloque actual, desplazado un valor de desviación. En algunos ejemplos, el codificador de vídeo 20/30 puede incluir un segundo IPMVC en la lista de candidatos, donde el segundo IPMVC es un vector de movimiento para un segundo bloque de referencia en la visualización de referencia a la que hace referencia el vector de disparidad sin desplazar el vector de disparidad.

- 5 **[0130]** El codificador de vídeo 20/30 puede comparar la información de movimiento del primer IPMVC y la información de movimiento del segundo IPMVC, como el vector de movimiento, el índice de imagen de referencia y la lista de imágenes de referencia. En algunos ejemplos, el codificador de vídeo 20/30 puede comparar la información de movimiento del primer IPMVC solo con la información de movimiento del segundo IPMVC, y sin otros candidatos en la lista de candidatos. Si el codificador de vídeo 20/30 determina que la información de movimiento del primer IPMVC y la información de movimiento del segundo IPMVC son iguales (por ejemplo, los vectores de movimiento son iguales o ambos vectores de movimiento y los índices de imagen de referencia son los mismos), el codificador de vídeo 20/30 puede determinar que el primer IPMVC no se incluirá en la lista de candidatos. Si el codificador de vídeo 20/30 determina que la información de movimiento del primer IPMVC y la información de movimiento del segundo IPMVC son diferentes (por ejemplo, los vectores de movimiento son diferentes, los vectores de movimiento son iguales, pero los índices de imagen de referencia son diferentes, los vectores de movimiento son diferentes, pero los índices de imagen de referencia son diferentes, o tanto los vectores de movimiento como los índices de imagen de referencia son diferentes), el codificador de vídeo 20/30 puede determinar que el primer IPMVC debe incluirse en la lista de candidatos.
- 15 **[0131]** En algunos ejemplos, el segundo IPMVC, del ejemplo anterior, siempre puede incluirse en la lista de candidatos. En tales ejemplos, la técnica de purga (por ejemplo, comparar el primer IPMVC y el segundo IPMVC y en algunos ejemplos solo con el segundo IPMVC) puede determinar si incluir o no el primer IPMVC en la lista de candidatos.
- 20 **[0132]** De acuerdo con otra técnica de esta divulgación, las técnicas anteriores solo pueden aplicarse cuando el vector de disparidad esté disponible después de NBDV. Si no se encuentra un vector de disparidad después de NBDV, el codificador de vídeo 20/30 puede no aplicar las técnicas anteriores para desplazar el vector de disparidad para obtener IPMVC e IDMVC adicionales (no se aplicará lo anterior). En un ejemplo, los IPMVC obtenidos a partir de los procedimientos anteriores siempre se aplican. Sin embargo, los IDMVC recién obtenidos se insertan en la lista de candidatos de fusión solo cuando el vector de disparidad está disponible con NBDV.
- 25 **[0133]** Además, en el modo de fusión y salto, el codificador de vídeo 20/30 puede obtener un IPMVC de un vector de disparidad, para un bloque actual, desplazado un primer valor de desviación, si el IPMVC está disponible, y obtener el IDMVC del vector de disparidad, para el bloque actual, desplazado un segundo valor de desviación diferente si se determina que el IPMVC no está disponible. Además, en el modo AMVP, el codificador de vídeo 20/30 puede obtener el IPMVC o el IDMVC basándose en el índice de imagen de referencia. En consecuencia, en algunos ejemplos, solo uno de los IPMVC o IDMVC se incluye en la lista de candidatos.
- 30 **[0134]** Las diversas técnicas introducidas anteriormente pueden implementarse mediante el codificador de vídeo 20/30. Ahora se analizarán varias implementaciones de ejemplo de las técnicas presentadas anteriormente. El vector de disparidad puede obtenerse de acuerdo con los procedimientos HEVC. Sin embargo, se pueden obtener más candidatos basándose en el vector de disparidad para los modos de salto/fusión o el modo AMVP.
- 35 **[0135]** De acuerdo con un aspecto de esta divulgación, el codificador de vídeo 20/30 puede configurarse para realizar la mutación del vector de disparidad para IPMVC e IDMVC. Basándose en el vector de disparidad, se utilizan dos mutaciones de vectores de disparidad para obtener IPMVC e IDMVC. La mutación del vector de disparidad desplazado hacia la izquierda o hacia la derecha (LDV y/o RDV) para la obtención de IPMVC se logra al aumentar el vector de disparidad en -16 y/o 16 en el componente horizontal. La mutación del vector de disparidad desplazado hacia la izquierda y/o hacia la derecha (LDV y/o RDV) para la obtención de IDMVC se logra al aumentar el vector de disparidad en -4 y/o 4 en el componente horizontal.
- 40 **[0136]** El codificador de vídeo 20/30 puede obtener IPMVC a partir de vectores de disparidad mutados. Con los vectores de disparidad desplazados, dos IPMVC adicionales pueden obtenerse de la misma manera que el IPMVC en el procedimiento actual. Los dos IPMVC recién obtenidos con LDV y RDV se indican mediante el IPMVC izquierdo y el IPMVC derecho, respectivamente. Para simplificar, los IPMVC (candidatos de movimiento temporal) obtenidos a partir de LDV y RDV se denominan LTMV y RTMV respectivamente.
- 45 **[0137]** El codificador de vídeo 20/30 puede obtener IDMVC a partir de vectores de disparidad mutados. Con los vectores de disparidad desplazados, dos IDMVC adicionales pueden obtenerse de la misma manera que el IDMVC en el procedimiento actual. Los dos IDMVC recién obtenidos con LDV y RDV se indican mediante IDMVC izquierdo y IDMVC derecho, respectivamente. Para simplificar, los IDMVC (candidatos de vector de movimiento de disparidad) obtenidos a partir de LDV y RDV se denominan LDMV y RDMV respectivamente. Basándose en un vector de disparidad mutado, LDV o RDV, se obtiene uno y solo un candidato, LTMV (RTMV) o LDMV (RDMV). Tal candidato se denota como LMV cuando se obtiene a partir de LDV, o RMV cuando se obtiene a partir de RDV. Un LMV se puede obtener de la siguiente manera: si LTMV está disponible: LMV se establece en LTMV; de lo contrario, LMV se establece en LDMV. Un RMV se puede obtener de la siguiente manera: si RTMV está disponible: RMV se establece en RTMV; de lo contrario, RMV se establece en RDMV.
- 50 **[0138]** El codificador de vídeo 20/30 puede aplicar el siguiente proceso de comprobación y purga después de que los IPMVC y los IDMVC se generen al cambiar los vectores de disparidad. El IPMVC izquierdo recién obtenido se compara
- 55
- 60
- 65

con el IPMVC obtenido con el vector de disparidad sin mutación, el candidato de fusión espacial de  $A_1$  o  $B_1$  en orden, si está disponible. Si es igual a cualquiera de estos tres candidatos, se eliminará de la lista de candidatos de fusión. De lo contrario, se agrega a la lista de candidatos de fusión después de todos los candidatos de fusión espacial, por lo tanto, antes del candidato de fusión temporal.

5 **[0139]** Si el IPMVC izquierdo no se agrega a la lista de candidatos de fusión, el IDMVC izquierdo recién obtenido se compara con el candidato de fusión espacial de  $A_1$  o  $B_1$  en orden, si está disponible. Si es igual a cualquiera de estos dos candidatos, se eliminará de la lista de candidatos de fusión. De lo contrario, se agrega a la lista de candidatos de fusión en la misma posición que el IPMVC izquierdo, si está disponible, es decir, después de todos los candidatos de fusión espacial, por lo tanto, antes del candidato de fusión temporal.

10 **[0140]** El nuevo IPMVC derecho recién obtenido se compara con el IPMVC obtenido con el vector de disparidad sin mutación, el candidato de fusión espacial de  $A_1$  o  $B_1$ , el IPMVC izquierdo en orden, si está disponible. Si es igual a cualquiera de estos cuatro candidatos, se eliminará de la lista de candidatos de fusión. De lo contrario, se agrega a la lista de candidatos que se fusionan después de todos los candidatos de fusión espacial y el IPMVC izquierdo o el IDMVC izquierdo.

15 **[0141]** Si el IPMVC correcto no se agrega a la lista de candidatos de fusión, el IDMVC derecho recién obtenido se compara con el candidato de fusión espacial de  $A_1$  o  $B_1$  en orden, si está disponible. Si es igual a cualquiera de estos dos candidatos, se eliminará de la lista de candidatos de fusión. De lo contrario, se agrega a la lista de candidatos de fusión en la misma posición que el IPMVC derecho, si está disponible, es decir, después de todos los candidatos de fusión espacial y el IPMVC izquierdo o el IDMVC izquierdo.

20 **[0142]** En una alternativa, solo cuando dos candidatos tienen el mismo tipo, es decir, son vectores de movimiento de disparidad o son vectores de movimiento temporal, se comparan. Por ejemplo, si el candidato obtenido a partir de  $A_1$  tiene una imagen de referencia temporal, no es necesaria la comparación entre este y el IDMVC izquierdo/derecho.

25 **[0143]** El codificador de vídeo 20/30 puede configurarse para implementar alternativas de varios aspectos descritos anteriormente. Por ejemplo, el codificador de vídeo 20/30 puede configurarse para determinar las posiciones de inserción para los nuevos candidatos obtenidos. En contraste con el ejemplo descrito anteriormente, donde los nuevos candidatos obtenidos se insertan después de que todos los candidatos de fusión espacial y el candidato (ya sea IPMVC o IDMVC) obtenido a partir de LDV preceda al obtenido a partir de RDV, se pueden aplicar las siguientes posiciones y órdenes:

- 35
- Después de un subconjunto de los candidatos de fusión espacial.
  - De forma alternativa, después de todos los candidatos espaciales, temporales y los candidatos obtenidos por el vector de disparidad, por lo tanto, antes de los candidatos combinados.
- 40
- De forma alternativa, el IPMVC obtenido con un valor desplazado igual a  $-4*M$ , si está disponible, se inserta antes del IDMVC obtenido con un valor desplazado igual a  $-4*N$ . El IPMVC obtenido con un valor desplazado igual a  $4*M$ , si está disponible, se inserta antes del IDMVC obtenido con un valor desplazado igual a  $4*N$ .
- 45
- De forma alternativa, todos los IPMVC obtenidos con valores desplazados se insertan en la lista, después del IPMVC obtenido a partir del vector de disparidad, por lo tanto, antes de todos los candidatos de fusión espacial.
  - De forma alternativa, todos los IDMVC obtenidos con valores desplazados se insertan en la lista, después del IDMVC obtenido a partir del vector de disparidad, por lo tanto, antes de los candidatos de fusión espacial obtenidos a partir de  $A_0$  y  $B_2$ .
- 50
- En una alternativa, los IPMVC y/o IDMVC recién agregados podrían insertarse en la lista de candidatos en cualquier posición y en cualquier orden.
  - Además, el candidato obtenido a partir de los RDV precede al obtenido a partir de los LDV.
- 55
- De forma alternativa, el candidato obtenido a partir de LDV y RDV no puede insertarse consecutivamente. En su lugar, el candidato obtenido a partir de LDV puede insertarse justo después del último candidato de fusión espacial y justo antes del candidato de fusión temporal, mientras que el candidato obtenido a partir de RDV puede insertarse justo después del candidato de fusión temporal, y viceversa.

60 **[0144]** El codificador de vídeo 20/30 puede configurarse para realizar un proceso de purga distinto al proceso de purga descrito anteriormente. Por ejemplo, el codificador de vídeo 20/30 puede configurarse para aplicar el siguiente proceso de purga.

- 65
- Solo uno o más candidatos espaciales selectivos (por ejemplo,  $A_1$ ) se comparan con los candidatos recién obtenidos.

- a. Por ejemplo, los candidatos recién obtenidos (LMV y/o RMV) se comparan solo con el candidato de fusión espacial de A1.
- 5 b. Por ejemplo, los candidatos recién obtenidos (LMV y/o RMV) se comparan solo con el candidato de fusión espacial de B1.
- De forma alternativa, los candidatos recién obtenidos no se comparan con ningún candidato espacial o temporal.
- 10 ◦ Además, los candidatos de vectores de movimiento recién agregados, excluyendo los obtenidos a partir del vector de disparidad, deben compararse entre sí para evitar duplicaciones.
- De forma alternativa, los candidatos de vectores de movimiento recién agregados, incluidos los obtenidos a partir del vector de disparidad, deben compararse entre sí para evitar duplicaciones.
- 15 ◦ De forma alternativa, los candidatos a vectores de movimiento recién agregados solo se comparan con los obtenidos a partir del vector de disparidad para evitar duplicaciones y no se comparan entre sí.
- En una alternativa, los IDMVC y/o IPMVC recién agregados deben compararse con todos o un subconjunto de candidatos de fusión disponibles.
  - De forma alternativa, además, si un candidato obtenido a partir de RDV o LDV (LMV y/o RMV) es un vector de movimiento de disparidad, nunca se compara con IDMVC.
- 25 ◦ En una alternativa, cuando los bloques de referencia ubicados con los vectores de disparidad desplazados ( $-4*M/4*M$ ) y el bloque de referencia ubicado con el vector de disparidad pertenecen a la misma PU, el proceso de purga entre los IPMVC recién agregados y el la obtenido a partir del vector de disparidad puede saltarse y los IPMVC recién agregados se eliminan de la lista de candidatos de fusión.
- 30 a. Además, en una alternativa, el IDMVC obtenido a partir de vectores de disparidad desplazados ( $-4*N/4*N$ ) no se agrega a la lista de candidatos de fusión.
- b. En otra alternativa, además, el IDMVC obtenido a partir de los vectores de disparidad desplazados ( $-4*N/4*N$ ) no se agrega a la lista de candidatos de fusión.
- 35 **[0145]** Como parte del proceso de purga, el codificador de vídeo 20/30 puede realizar una o más comparaciones como se introdujo anteriormente. En un ejemplo, se comparan los siguientes (hasta cinco) pares de candidatos de vectores de movimiento (si ambos están disponibles):
- 40 Cuando LTMV está disponible, (LMV, IPMVC obtenido a partir del vector de disparidad);  
(LMV, candidato de fusión espacial de B<sub>1</sub>);
- 45 Cuando RTMV está disponible, (RMV, IPMVC obtenido a partir del vector de disparidad);  
(RMV, candidato de fusión espacial de B<sub>1</sub>);
- Cuando tanto LTMV como LTMV están disponibles, (RMV, LMV)
- 50 Si se da una condición en un par de candidatos dado, por ejemplo, "Cuando LTMV está disponible", significa que el par correspondiente no se compara si la condición no es verdadera.
- [0146]** Tenga en cuenta que para cada par de candidatos (C0, C1), si son diferentes, C0 se inserta en la lista de candidatos de fusión, de lo contrario, C0 no se inserta más en la lista de candidatos de fusión. Además, si C1 no está en la lista de candidatos de fusión, se evita la comparación. Supongamos que los índices de referencia y los vectores de movimiento de C0 y C1 son RefldxLXC0, mvLXC0 y RefldxLXC1, mvLXC1 respectivamente, con X igual a 0 y 1. C0 y C1 se consideran diferentes si una o más de las siguientes condiciones son verdaderas:
- 60 Para la indicación de la lista de predicción X es 0 y 1,
- RefldxLXC0 no es igual a RefldxLXC1;
  - mvLXC0[0] no es igual a mvLXC1[0] cuando RefldxLXC0 no es igual a -1;
  - mvLXC0[1] no es igual a mvLXC1[1] cuando RefldxLXC0 no es igual a -1;
- 65

Tenga en cuenta que si un vector de movimiento no está disponible para una indicación de lista de predicción X determinada, el índice de referencia correspondiente puede considerarse igual a -1.

5 **[0147]** En un segundo ejemplo, se comparan los siguientes (hasta 2) pares de candidatos de vectores de movimiento (si ambos están disponibles):

    Cuando LTMV está disponible, (LMV, IPMVC obtenido a partir del vector de disparidad)

10      Cuando RTMV está disponible, (RMV, IPMVC obtenido a partir del vector de disparidad).

**[0148]** En un tercer ejemplo, se comparan los siguientes pares de candidatos de vectores de movimiento (si ambos están disponibles):

15      Cuando LTMV está disponible, (LMV, IPMVC obtenido a partir del vector de disparidad);

    (LMV, candidato de fusión espacial de  $B_1$ );

    Cuando RTMV está disponible, (RMV, IPMVC obtenido a partir del vector de disparidad)

20      (RMV, candidato de fusión espacial de  $B_1$ ).

**[0149]** En un cuarto ejemplo, se comparan los siguientes pares de candidatos de vectores de movimiento (si ambos están disponibles):

25      Cuando LTMV está disponible, (LMV, IPMVC obtenido a partir del vector de disparidad);

    (LMV, candidato de fusión espacial de  $A_1$ );

30      Cuando RTMV está disponible, (RMV, IPMVC obtenido a partir del vector de disparidad)

    (RMV, candidato de fusión espacial de  $A_1$ )

**[0150]** En un quinto ejemplo, se comparan los siguientes (hasta cinco) pares de candidatos de vectores de movimiento (si ambos están disponibles):

35      Cuando LTMV está disponible, (LMV, IPMVC obtenido a partir del vector de disparidad);

    (LMV, candidato de fusión espacial de  $A_1$ );

40      Cuando RTMV está disponible, (RMV, IPMVC obtenido a partir del vector de disparidad);

    (RMV, candidato de fusión espacial de  $A_1$ );

45      Cuando tanto LTMV como LTMV están disponibles, (RMV, LMV)

**[0151]** La FIG. 7 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo 20 que puede implementar las técnicas descritas en esta divulgación, incluyendo las técnicas de predicción entre visualizaciones descritas anteriormente. El codificador de vídeo 20 puede realizar la intracodificación y la intercodificación de bloques de vídeo dentro de fragmentos de vídeo. La intercodificación se puede denominar codificación mediante predicción intra, y la intercodificación se puede denominar codificación mediante predicción inter, cuando se realiza mediante el codificador de vídeo 20. La intracodificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial en el vídeo dentro de una trama o imagen de vídeo dada. La intercodificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal en el vídeo dentro de tramas o imágenes adyacentes de una secuencia de vídeo. La intercodificación también puede referirse a la codificación de visualización inter donde una imagen en otra visualización se usa como imagen de referencia para codificar un bloque actual en una imagen actual en una visualización actual. El modo intra (modo I) puede referirse a cualquiera de varios modos de compresión espacial. Los modos inter, tales como la predicción unidireccional (modo P) o la predicción bidireccional (modo B), pueden referirse a cualquiera de varios modos de compresión temporales o modos de compresión entre visualizaciones.

60 **[0152]** En el ejemplo de la FIG. 7, el codificador de vídeo 20 incluye una unidad de división 35, una unidad de procesamiento de predicción 41, una unidad de filtro 63, una memoria de imágenes de referencia 64, un sumador 50, una unidad de transformada 52, una unidad de cuantificación 54 y una unidad de codificación por entropía 56. La unidad de procesamiento de predicción 41 incluye la unidad de estimación de movimiento y disparidad 42, la unidad de compensación de movimiento y disparidad 44 y la unidad de procesamiento de predicción intra 46. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 incluye también una unidad de cuantificación inversa 58, una unidad de procesamiento de transformada inversa 60 y un sumador 62. La unidad de filtro 63 está destinada

a representar uno o más filtros de bucle tales como un filtro de eliminación de bloques, un filtro de bucle adaptativo (ALF) y un filtro de desplazamiento adaptativo de muestras (SAO). Aunque la unidad de filtro 63 que se muestra en la FIG. 7 es un filtro de bucle, en otras configuraciones, la unidad de filtro 63 puede implementarse como un filtro post-bucle.

5 **[0153]** Como se representa en la FIG. 7, el codificador de vídeo 20 recibe datos de vídeo, y la unidad de división 35 divide los datos en bloques de vídeo. Esta división también puede incluir la división en fragmentos, mosaicos u otras unidades mayores, así como la división de bloques de vídeo, por ejemplo, de acuerdo con una estructura de árbol cuaternario de LCU y CU. El codificador de vídeo 20 ilustra, en general, los componentes que codifican bloques de vídeo dentro de un fragmento de vídeo que se va a codificar. El fragmento se puede dividir en múltiples bloques de vídeo (y, posiblemente, en conjuntos de bloques de vídeo denominados mosaicos). La unidad de procesamiento de predicción 41 puede seleccionar uno entre una pluralidad de posibles modos de codificación, tal como uno entre una pluralidad de modos de intracodificación, o uno entre una pluralidad de modos de intercodificación, para el bloque de vídeo actual basándose en resultados de errores (por ejemplo, la velocidad de codificación y el nivel de distorsión). La unidad de procesamiento de predicción 41 puede proporcionar 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 imagen de referencia.

20 **[0154]** La unidad de procesamiento de predicción intra 46, dentro de la unidad de procesamiento de predicción 41, puede realizar la codificación con predicción intra del bloque de vídeo actual con respecto a uno o más bloques contiguos en la misma trama o fragmento que el bloque que vaya a codificarse, para proporcionar compresión espacial. La unidad de estimación de movimiento y disparidad 42 y la unidad de compensación de movimiento y disparidad 44 de la unidad de procesamiento de predicción 41 realizan la codificación mediante predicción inter y/o la codificación entre visualizaciones del bloque de vídeo actual en relación con uno o más bloques predictivos de una o más imágenes de referencia y/o visualizaciones de referencia, para proporcionar compresión temporal o de visualizaciones.

30 **[0155]** La unidad de estimación de movimiento y disparidad 42 puede estar configurada para determinar el modo de predicción inter y/o el modo de predicción entre visualizaciones para un fragmento de vídeo de acuerdo con un patrón predeterminado para una secuencia de vídeo. El patrón predeterminado puede designar fragmentos de vídeo de la secuencia como fragmentos P o fragmentos B. La unidad de estimación de movimiento y disparidad 42 y la unidad de compensación de movimiento y disparidad 44 pueden estar sumamente integradas, pero se ilustran por separado con fines conceptuales. La estimación de movimiento y disparidad, llevada a cabo por la unidad de estimación de movimiento y disparidad 42, es el proceso de generar vectores de movimiento, que estiman el movimiento para los bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de una PU de un bloque de vídeo de una trama o imagen de vídeo actual en relación con un bloque predictivo de una imagen de referencia. La estimación de disparidad, realizada por la unidad de estimación de movimiento y disparidad 42, es el proceso de generación de vectores de movimiento de disparidad, que puede usarse para predecir un bloque codificado actualmente a partir de un bloque de una visualización diferente.

40 **[0156]** Un bloque predictivo es un bloque que resulta corresponder estrechamente con la PU del bloque de vídeo que se va a codificar en términos de diferencia de píxel, que puede determinarse mediante una suma de una 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 fraccionarias de píxeles de imágenes de referencia almacenadas en la memoria de imágenes de referencia 64. Por ejemplo, el codificador de vídeo 20 puede interpolar valores de posiciones de un cuarto de píxel, posiciones de un octavo de píxel u otras posiciones fraccionarias de píxel de la imagen de referencia. Por lo tanto, la unidad de estimación de movimiento y disparidad 42 puede realizar una búsqueda de movimiento en relación con las posiciones de píxeles completas y las posiciones de píxeles fraccionarias, y emitir un vector de movimiento con una precisión de píxel fraccionaria.

50 **[0157]** La unidad de estimación de movimiento y disparidad 42 calcula un vector de movimiento (para predicción con compensación de movimiento) y/o un vector de movimiento de disparidad (para predicción con compensación de disparidad) para una PU de un bloque de vídeo en un fragmento sometido a intercodificación o predicción entre visualizaciones, 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 entre una primera lista de imágenes de referencia (RefPicList0) o una segunda lista de imágenes de referencia (RefPicList1), 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 y disparidad 42 envía el vector de movimiento y/o el vector de disparidad calculados a la unidad de codificación por entropía 56 y la unidad de compensación de movimiento 44.

60 **[0158]** La compensación de movimiento y/o la compensación de disparidad, realizadas por la unidad de compensación de movimiento y disparidad 44, pueden implicar obtener o generar el bloque predictivo basándose en el vector de movimiento determinado mediante estimación de movimiento y/o estimación de paridad, realizando posiblemente interpolaciones hasta la precisión de subpíxel. Tras recibir el vector de movimiento y/o la disparidad para la PU del bloque de vídeo actual, la unidad de compensación de movimiento y disparidad 44 puede localizar el bloque predictivo al que apunta el vector de movimiento y/o el vector de disparidad en una de las listas de imágenes de referencia. El codificador de vídeo 20 forma un bloque de vídeo residual restando los 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 píxel. Los valores de diferencia de píxel forman datos residuales para el bloque, y pueden incluir componentes de diferencia tanto de luma como de croma. El sumador 50 representa el componente o los componentes que realizan esta operación de resta. La unidad de compensación de movimiento y disparidad 44 también puede generar elementos sintácticos asociados a los bloques de vídeo y al fragmento de vídeo para su uso por el descodificador de vídeo 30 en la descodificación de los bloques de vídeo del fragmento de vídeo.

**[0159]** La unidad de procesamiento de predicción intra 46 puede realizar predicción intra de un bloque actual, de forma alternativa a la predicción inter llevada a cabo por la unidad de estimación de movimiento y disparidad 42 y la unidad de compensación de movimiento y disparidad 44, como se ha descrito anteriormente. En particular, la unidad de procesamiento de predicción intra 46 puede determinar un modo de predicción intra que se vaya a usar para codificar un bloque actual. En algunos ejemplos, la unidad de procesamiento de predicción intra 46 puede codificar un bloque actual usando diversos modos de predicción intra, por ejemplo, durante pases de codificación independientes, y la unidad de procesamiento de predicción intra 46 puede seleccionar un modo de predicción intra adecuado para su uso a partir de los modos probados. Por ejemplo, la unidad de procesamiento de predicción intra 46 puede calcular valores de velocidad-distorsión usando un análisis de velocidad-distorsión para las diversos modos de predicción intra probados, y seleccionar el modo de predicción intra que tenga las mejores características de velocidad-distorsión entre los modos probados. El análisis de velocidad-distorsión determina, en general, una magnitud de distorsión (o error) entre un bloque codificado y un bloque original no codificado que fue 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 procesamiento de predicción intra 46 puede calcular razones a partir de las distorsiones y velocidades para los diversos bloques codificados, para determinar qué modo de predicción intra presenta el mejor valor de velocidad-distorsión para el bloque.

**[0160]** En cualquier caso, después de seleccionar un modo de predicción intra para un bloque, la unidad de procesamiento de predicción intra 46 puede proporcionar información indicativa del modo de predicción intra seleccionado para el bloque a la unidad de codificación por entropía 56. La unidad de codificación por entropía 56 puede codificar la información que indique el modo de predicción intra seleccionado de acuerdo con las técnicas de esta divulgación. El codificador de vídeo 20 puede incluir, en el flujo de bits transmitido, datos de configuración, que pueden incluir una pluralidad de tablas de índices de modo de predicción intra y una pluralidad de tablas modificadas de índices de modo de predicción intra (también denominadas tablas de correlación de palabras de código), definiciones de contextos de codificación para diversos bloques e indicaciones de un modo de predicción intra más probable, una tabla de índices de modo de predicción intra y una tabla modificada de índices de modo de predicción intra, a usar para cada uno de los contextos.

**[0161]** Después de que la unidad de procesamiento de predicción 41 genera el bloque predictivo para el bloque de vídeo actual, ya sea mediante la predicción inter o la predicción intra, el codificador de vídeo 20 forma un bloque de vídeo residual restando el bloque predictivo al bloque de vídeo actual. Los datos de vídeo residuales en el bloque residual pueden ser incluidos en una o más TU y aplicados a la unidad de transformada 52. La unidad de transformada 52 transforma los datos de vídeo residuales en coeficientes de transformada residual, usando una transformada, tal como una transformada de coseno discreta (DCT) o una transformada conceptualmente similar. La unidad de transformada 52 puede convertir los datos de vídeo residuales de un dominio de píxeles a un dominio de transformada, tal como un dominio de frecuencia.

**[0162]** La unidad 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 más la velocidad de bits. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos, o a la totalidad, de los coeficientes. El grado de cuantificación se puede modificar ajustando un parámetro de cuantificación. En algunos ejemplos, la unidad 54 de cuantificación puede realizar, a continuación, una exploración de la matriz que incluye los coeficientes de transformada cuantificados. De forma alternativa, la unidad de codificación por entropía 56 puede realizar la exploración.

**[0163]** Tras la cuantificación, la unidad de codificación por entropía 56 realiza la codificación por entropía de los coeficientes de transformada cuantificados. Por ejemplo, la unidad de codificación por entropía 56 puede realizar una codificación de longitud variable adaptativa al contexto (CAVLC), una codificación aritmética binaria adaptativa al contexto (CABAC), una codificación aritmética binaria adaptativa al contexto basada en la sintaxis (SBAC), una codificación por entropía por división de intervalos de probabilidad (PIPE) u otros procedimientos o técnicas de codificación por entropía. Tras la codificación por entropía por la unidad de codificación por entropía 56, el flujo de bits codificado se puede transmitir al descodificador de vídeo 30, o archivarse para su posterior transmisión o recuperación por el descodificador de vídeo 30. La unidad de codificación por entropía 56 también puede realizar la codificación por entropía de los vectores de movimiento y los otros elementos sintácticos para el fragmento de vídeo actual que se está codificando.

**[0164]** La unidad de cuantificación inversa 58 y la unidad de procesamiento de transformada inversa 60 aplican una cuantificación inversa y una transformación inversa, respectivamente, para reconstruir el bloque residual en el dominio del píxel, para su posterior uso como bloque de referencia de una imagen de referencia. La unidad de compensación

de movimiento y disparidad 44 puede calcular un bloque de referencia sumando el bloque residual a un bloque predictivo de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. La unidad de compensación de movimiento y disparidad 44 también puede aplicar uno o más filtros de interpolación al bloque residual reconstruido para calcular valores de píxeles fraccionarios, para su uso en la estimación de movimiento. El sumador 62 añade el bloque residual reconstruido al bloque de predicción compensado en movimiento, generado por la unidad de compensación de movimiento y disparidad 44 para generar un bloque de referencia para su almacenamiento en la memoria de imágenes de referencia 64. La unidad de estimación de movimiento y disparidad 42 y la unidad de compensación de movimiento y disparidad 44 pueden usar el bloque de referencia como un bloque de referencia para realizar predicción inter de un bloque en una posterior imagen o trama de vídeo.

[0165] La FIG. 8 es un diagrama de bloques que ilustra un descodificador de vídeo de ejemplo 30 que puede implementar las técnicas descritas en esta divulgación, incluidas las técnicas de predicción entre visualizaciones descritas anteriormente. En el ejemplo de la FIG. 8, el descodificador de vídeo 30 incluye una unidad de descodificación por entropía 80, una unidad de procesamiento de predicción 81, una unidad de cuantificación inversa 86, una unidad de procesamiento de transformación inversa 88, un sumador 90, una unidad de filtro 91 y una memoria de imágenes de referencia 92. La unidad de procesamiento de predicción 81 incluye la unidad de compensación de movimiento y disparidad 82 y la unidad de procesamiento de predicción intra 84. En algunos ejemplos, el descodificador de vídeo 30 puede realizar una pasada de descodificación que en general es recíproca a la pasada de codificación descrita con respecto al codificador de vídeo 20 de la FIG. 7.

[0166] Durante el proceso de descodificación, el descodificador de vídeo 30 recibe un flujo de bits de vídeo codificado, que representa bloques de vídeo de un fragmento de vídeo codificado y elementos sintácticos asociados, desde el codificador de vídeo 20. La unidad de descodificación por entropía 80 del descodificador de vídeo 30 descodifica por entropía el flujo de bits para generar coeficientes cuantificados, vectores de movimiento y otros elementos sintácticos. La unidad de descodificación por entropía 80 envía los vectores de movimiento y otros elementos sintácticos a la unidad de procesamiento de predicción 81. El descodificador de vídeo 30 puede recibir los elementos sintácticos en el nivel de fragmento de vídeo y/o el nivel de bloque de vídeo.

[0167] Cuando el fragmento de vídeo se codifica como un fragmento intracodificado (I), la unidad de procesamiento de predicción intra 84 de la unidad de procesamiento de predicción 81 puede generar datos de predicción para un bloque de vídeo del fragmento de vídeo actual basándose en un modo de predicción intra señalado, y datos de bloques previamente descodificados de la trama o imagen actual. Cuando la trama de vídeo se codifica como un fragmento intercodificado (es decir, B, P o GPB), la unidad de compensación de movimiento y disparidad 82 de la unidad de procesamiento de predicción 81 genera bloques predictivos para un bloque de vídeo del fragmento de vídeo actual, basándose en los vectores de movimiento y otros elementos sintácticos recibidos desde la unidad de descodificación por entropía 80. Los bloques predictivos se pueden generar a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El descodificador de vídeo 30 puede construir las listas de tramas de referencia, lista 0 y lista 1, mediante técnicas de construcción predeterminadas, basándose en imágenes de referencia almacenadas en la memoria de imágenes de referencia 92.

[0168] La unidad de compensación de movimiento y disparidad 82 determina la información de predicción para un bloque de vídeo del fragmento de vídeo actual, analizando sintácticamente los vectores de movimiento y otros elementos sintácticos, y usa la información de predicción para generar los bloques predictivos para el bloque de vídeo actual que está siendo descodificado. Por ejemplo, la unidad de compensación de movimiento y disparidad 82 utiliza algunos de los elementos sintácticos recibidos para determinar un modo de predicción (por ejemplo, predicción intra o predicción inter) utilizado para codificar los bloques de vídeo del fragmento de vídeo, un tipo de fragmento de predicción inter o predicción entre visualizaciones (por ejemplo, fragmento B o fragmento P), información de construcción para una o más de las listas de imágenes de referencia para el fragmento, vectores de movimiento y/o vectores de disparidad para cada bloque de vídeo intercodificado del fragmento, estado de predicción inter para cada bloque de vídeo intercodificado del fragmento y otra información para descodificar los bloques de vídeo del fragmento de vídeo actual.

[0169] En algunos ejemplos, la unidad de compensación de movimiento y disparidad 82 puede determinar los elementos sintácticos señalados que indican vectores de movimiento usando un proceso de predicción de vector de movimiento. Los procesos de predicción del vector de movimiento pueden incluir un modo AMVP y un modo de fusión. La unidad de compensación de movimiento y disparidad 82 también puede realizar la interpolación basándose en filtros de interpolación. La unidad de compensación de movimiento y disparidad 82 puede usar filtros de interpolación como los usados por el codificador de vídeo 20 durante la codificación de los bloques de vídeo, para calcular valores interpolados para píxeles fraccionarios de bloques de referencia. En este caso, la unidad de compensación de movimiento y disparidad 82 puede determinar los filtros de interpolación utilizados por el codificador de vídeo 20 a partir de los elementos sintácticos recibidos, y utilizar los filtros de interpolación para generar bloques predictivos.

[0170] La unidad de cuantificación inversa 86 realiza la cuantificación inversa, es decir, la decuantificación, de los coeficientes de transformada cuantificados proporcionados en el flujo de bits y descodificados por la unidad de descodificación por entropía 80. El proceso de cuantificación inversa puede incluir el uso de un parámetro de cuantificación calculado por el codificador de vídeo 20 para cada bloque de vídeo del fragmento de vídeo con el fin de

determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa que se debería aplicar. La unidad de procesamiento de transformada inversa 88 aplica una transformada inversa, por ejemplo una DCT inversa, una transformada inversa de enteros o un proceso de transformada inversa conceptualmente similar, a los coeficientes de transformada, con el fin de generar bloques residuales en el dominio del píxel.

5 **[0171]** Después de que la unidad de compensación de movimiento y disparidad 82 genera el bloque predictivo para el bloque de vídeo actual, basándose en los vectores de movimiento y otros elementos sintácticos, el descodificador de vídeo 30 forma un bloque de vídeo descodificado sumando los bloques residuales procedentes de la unidad de procesamiento de transformada inversa 88 a los correspondientes bloques predictivos generados por la unidad de  
10 compensación de movimiento y disparidad 82. El sumador 90 representa el componente o los componentes que realizan esta operación de suma. Si se desea, también pueden usarse filtros de bucle (ya sea en el bucle de codificación o después del bucle de codificación) para suavizar las transiciones de píxeles o mejorar de otro modo la calidad del vídeo. La unidad de filtro 91 está destinada a representar uno o más filtros de bucle tales como un filtro de eliminación de bloques, un filtro de bucle adaptativo (ALF) y un filtro de desplazamiento adaptativo de muestras (SAO).  
15 Aunque la unidad de filtro 91 que se muestra en la FIG. 8 es un filtro de bucle, en otras configuraciones, la unidad de filtro 91 puede implementarse como un filtro post-bucle. Los bloques de vídeo descodificados de una trama o imagen determinada se almacenan a continuación en la memoria de imágenes de referencia 92, que almacena imágenes de referencia usadas para una subsiguiente compensación de movimiento. La memoria de imágenes de referencia 92 almacena también vídeo descodificado para su presentación posterior en un dispositivo de visualización, tal como el  
20 dispositivo de visualización 32 de la FIG. 1.

**[0172]** La FIG. 9 es un diagrama de flujo que muestra un proceso de codificación a modo de ejemplo de acuerdo con las técnicas de la divulgación. Las técnicas de la FIG. 9 se describirán con respecto al codificador de vídeo 20, pero  
25 pueden ser realizadas por otros tipos de codificadores de vídeo. El codificador de vídeo 20 determina si un bloque actual tiene predicción inter en el modo de fusión, modo de salto o modo AMVP (180).

**[0173]** Si el bloque actual tiene predicción inter en el modo AMVP, el codificador de vídeo 20 puede determinar si obtener un IDMVC o un IPMVC basándose en un índice de imagen de referencia (182). Por ejemplo, si el índice de imagen de referencia se refiere a una imagen de referencia temporal, el codificador de vídeo 20 puede determinar un  
30 IPMVC, y si el índice de imagen de referencia se refiere a una imagen de referencia entre visualizaciones, el codificador de vídeo 20 puede determinar un IDMVC.

**[0174]** Si el bloque actual tiene predicción inter en el modo de fusión o en el modo de salto, el codificador de vídeo 20 puede determinar la disponibilidad del IPMVC (184). El codificador de vídeo 20 puede determinar que el IPMVC está disponible si la información de movimiento para un bloque de referencia al que hace referencia mediante un vector de  
35 disparidad, del bloque actual, desplazado un valor de desviación, incluye un vector de movimiento que es un vector de movimiento temporal que se refiere a una imagen en la visualización de referencia con un valor POC igual a un valor POC de una imagen identificada en una lista de imágenes de referencia del bloque actual. Como se describió anteriormente, el IPMVC puede ser información de movimiento de un bloque de referencia en una visualización de referencia a la que se hace referencia mediante un vector de disparidad, para el bloque actual, desplazado un primer  
40 valor de desviación, y el IDMVC puede ser un vector de disparidad, para el bloque actual, desplazado un segundo valor de desviación diferente.

**[0175]** En algunos ejemplos, si IPMVC está disponible, el codificador de vídeo 20 puede obtener el IPMVC, y evitar la obtención del IDMVC en modo de fusión o salto. Si IPMVC no está disponible y el bloque actual tiene predicción inter en el modo de fusión o en el modo de salto, el codificador de vídeo 20 puede obtener el IDMVC.  
45

**[0176]** El codificador de vídeo 20 puede determinar si el IPMVC o el IDMVC se incluirán en una lista de candidatos (186). En algunos ejemplos, el codificador de vídeo 20 puede incluir solo uno de los IPMVC y los IDMVC, y no ambos (es decir, solo uno de los IPMVC y los IDMVC están incluidos en la lista de candidatos). Por ejemplo, en el modo de fusión, si el codificador de vídeo 20 obtiene el IPMVC porque el IPMVC está disponible, el codificador de vídeo 20 puede evitar la obtención del IDMVC. En el modo AMVP, el codificador de vídeo 20 puede incluir uno de los IPMVC o IDMVC en una lista de candidatos basándose en si el índice de la imagen de referencia se refiere a una imagen de  
50 referencia temporal o una imagen de referencia entre visualizaciones.

**[0177]** Además, en algún ejemplo, el codificador de vídeo 20 puede obtener otro IPMVC basado en el vector de disparidad sin desplazar el vector de disparidad. En estos ejemplos, el codificador de vídeo 20 puede comparar la información de movimiento para los dos IPMVC (es decir, uno basado en el vector de disparidad desplazado y uno basado en el vector de disparidad no desplazado). En algunos ejemplos, el codificador de vídeo 20 puede comparar la información de movimiento para el IPMVC obtenido a partir del desplazamiento del vector de disparidad solo con la información de movimiento para el IPMVC obtenido sin desplazar el vector de disparidad. Si la información de movimiento de los dos IPMVC es la misma, el codificador de vídeo 20 puede no incluir el IPMVC basándose en el vector de disparidad desplazado en la lista de candidatos, y si la información de movimiento de los dos IPMVC es diferente, el codificador de vídeo 20 puede incluir el IPMVC basándose en el vector de disparidad desplazado en la lista de candidatos. El codificador de vídeo 20 puede incluir siempre el IPMVC basado en el vector de disparidad sin desplazamiento.  
55  
60  
65

**[0178]** El codificador de vídeo 20 puede codificar mediante predicción inter el bloque actual basándose en la lista de candidatos (188). El codificador de vídeo 20 también puede señalar elementos sintácticos (por ejemplo, elementos sintácticos de señalización) que indican si el bloque actual tiene predicción inter en el modo de fusión, modo de salto o modo AMVP y un índice en la lista de candidatos que el descodificador de vídeo 30 usa para determinar el vector de movimiento para el bloque actual para codificación mediante predicción inter del bloque actual (190).

**[0179]** La FIG. 10 es un diagrama de flujo que muestra un proceso de descodificación de ejemplo de acuerdo con las técnicas de la divulgación. Las técnicas de la FIG. 10 se describirán con respecto al descodificador de vídeo 30 pero pueden ser realizadas por otros tipos de descodificador de vídeo. El descodificador de vídeo 30 recibe e indexa en una lista de candidatos (200). El descodificador de vídeo 30 puede recibir elementos sintácticos (por ejemplo, elementos sintácticos de descodificación o elementos sintácticos de descodificación) que indican si un bloque actual tiene predicción inter en modo de fusión, modo de salto o modo AMVP para determinar si el bloque actual tiene predicción inter en modo de fusión, modo de salto o modo AMVP (202).

**[0180]** Si el descodificador de vídeo 30 determina que el bloque actual tiene predicción inter en el modo AMVP, el descodificador de vídeo 30 puede determinar si obtener un IDMVC o un IPMVC basándose en un índice de imagen de referencia (204). Por ejemplo, si el índice de imagen de referencia se refiere a una imagen de referencia temporal, el descodificador de vídeo 30 puede determinar un IPMVC, y si el índice de imagen de referencia se refiere a una imagen de referencia entre visualizaciones, el descodificador de vídeo 30 puede determinar un IDMVC.

**[0181]** Si el bloque actual tiene predicción inter en el modo de fusión o en el modo de salto, el descodificador de vídeo 30 puede determinar la disponibilidad del IPMVC (206). El descodificador de vídeo 30 puede determinar que el IPMVC está disponible si la información de movimiento para un bloque de referencia al que se hace referencia mediante un vector de disparidad, del bloque actual, desplazado un valor de desviación incluye un vector de movimiento que es un vector de movimiento temporal que se refiere a una imagen en la visualización de referencia con un valor POC igual a un valor POC de una imagen identificada en una lista de imágenes de referencia del bloque actual. De nuevo, como se describió anteriormente, el IPMVC puede ser un vector de movimiento para un bloque de referencia en una visualización de referencia a la que se hace referencia mediante un vector de disparidad, para el bloque actual, desplazado un primer valor de desviación, y el IDMVC puede ser un vector de disparidad, para el bloque actual, desplazado un segundo valor de desviación diferente.

**[0182]** En algunos ejemplos, si IPMVC está disponible, el descodificador de vídeo 30 puede obtener el IPMVC, y evitar la obtención del IDMVC en el modo de fusión o salto. Si IPMVC no está disponible y el bloque actual tiene predicción inter en el modo de fusión o en el modo de salto, el descodificador de vídeo 30 puede obtener el IDMVC.

**[0183]** El descodificador de vídeo 30 puede determinar si el IPMVC o el IDMVC tiene que incluir en una lista de candidatos (208). En algunos ejemplos, el descodificador de vídeo 30 puede incluir solo uno de los IPMVC y los IDMVC, y no ambos (es decir, solo uno de los IPMVC y los IDMVC están incluidos en la lista de candidatos). Por ejemplo, en el modo de fusión, si el descodificador de vídeo 30 obtiene el IPMVC porque el IPMVC está disponible, el descodificador de vídeo 30 puede evitar la obtención del IDMVC. En el modo AMVP, el descodificador de vídeo 30 puede incluir uno de los IPMVC o IDMVC en una lista de candidatos basándose en si el índice de la imagen de referencia se refiere a una imagen de referencia temporal o una imagen de referencia entre visualizaciones. El descodificador de vídeo 30 puede descodificar mediante predicción inter el bloque actual basándose en la lista de candidatos (210).

**[0184]** Además, en algunos ejemplos, el descodificador de vídeo 30 puede obtener otro IPMVC basado en el vector de disparidad sin desplazar el vector de disparidad. En estos ejemplos, el descodificador de vídeo 30 puede comparar la información de movimiento para los dos IPMVC (es decir, uno basado en el vector de disparidad desplazado y uno basado en el vector de disparidad sin desplazar). En algunos ejemplos, el descodificador de vídeo 30 puede comparar la información de movimiento para el IPMVC obtenido a partir del desplazamiento del vector de disparidad solo con la información de movimiento para el IPMVC obtenido sin desplazar el vector de disparidad. Si la información de movimiento de los dos IPMVC es la misma, es posible que el descodificador de vídeo 30 no incluya el IPMVC basado en el vector de disparidad desplazado en la lista de candidatos, y si la información de movimiento de los dos IPMVC es diferente, el descodificador de vídeo 30 puede incluir el IPMVC basándose en el vector de disparidad desplazado en la lista de candidatos. El descodificador de vídeo 30 puede incluir siempre el IPMVC basado en el vector de disparidad sin desplazamiento.

**[0185]** Además, aunque las técnicas de ejemplo anteriores se describen con respecto a la codificación de vídeo de múltiples visualizaciones, las técnicas descritas en esta divulgación no están necesariamente limitadas. Por ejemplo, las técnicas descritas en esta divulgación pueden extenderse a la codificación de vídeo escalable (SVC), incluidas las extensiones a la norma HEVC para la codificación de vídeo escalable (SHEVC).

**[0186]** En la codificación de vídeo escalable (SVC), podría haber múltiples capas. La capa en el nivel más bajo puede servir simplemente como una capa base (BL) y la capa en el nivel más alto como una capa de mejora (EL). Todas las capas en el medio pueden servir como EL y BL. Por ejemplo, una capa en el medio puede ser una EL para las capas

debajo de ella, y al mismo tiempo como una BL para las capas que están por encima de ella. Para simplificar la descripción, la divulgación supone que existen dos capas, una BL y una EL, para ilustrar las técnicas actuales. Cabe señalar que las técnicas descritas en esta divulgación también son aplicables a casos con múltiples capas también.

5 **[0187]** El concepto de un vector de disparidad en SCV puede referirse a un vector que se refiere a un bloque de referencia en una capa de referencia, donde la capa de referencia es otra capa distinta a la capa que incluye el bloque del cual se realiza predicción inter. Un ejemplo de dicho vector entre capas puede ser un vector cero (por ejemplo, un vector con componentes (0, 0)). Para SCV, el vector entre capas se puede utilizar de una manera similar a la descrita anteriormente para el vector de disparidad con respecto al vídeo 3D.

10 **[0188]** Como se describió anteriormente, para el vídeo 3D, el codificador de vídeo 20/30 puede obtener un IPMVC y un IDMVC sin desplazar el vector de disparidad y obtener los IPMVC e IDMVC adicionales al desplazar el vector de disparidad, y en algunos ejemplos, en diferentes cantidades (es decir, diferentes valores de desplazamiento) para los IPMVC adicionales en relación con los IDMVC adicionales, a los que se hace referencia en general como candidatos entre visualizaciones. Para SVC, el codificador de vídeo 20/30 también puede obtener candidatos entre capas. Como ejemplo, el codificador de vídeo 20/30 puede obtener un candidato entre capas basado en información de movimiento heredada de un bloque de referencia en la capa de referencia a la que se refiere el vector entre capas y/o basado en información de movimiento heredada de un bloque de referencia en la capa de referencia a la que se refiere el vector entre capas, desplazado un valor de desviación. Como otro ejemplo, el codificador de vídeo 20/30 puede obtener un candidato entre capas convirtiendo el vector entre capas en el candidato entre capas y/o convirtiendo el vector entre capas, desplazado un valor de desviación, en el intercambio entre capas candidato entre capas.

20 **[0189]** En consecuencia, en algunos ejemplos, las técnicas descritas en esta divulgación para obtener el IPMVC y el IDMVC, y las técnicas para cuando se obtiene el IPMVC y el IDMVC (por ejemplo, en el modo de fusión, el modo de salto o el modo AMVP) pueden extenderse a SVC. Para SVC, el codificador de vídeo 20/30 puede obtener un primer candidato entre capas de una manera similar a obtener el IPMVC (por ejemplo, basado en información de movimiento para un bloque de referencia en una capa de referencia a la cual hace referencia un vector entre capas o un vector entre capas desplazado para el bloque actual) y puede obtener un segundo candidato entre capas de manera similar a obtener el IDMVC (por ejemplo, convertir el vector entre capas o el vector entre capas desplazado para el bloque actual en el candidato entre capas). Para SVC, el codificador de vídeo 20/30 puede implementar técnicas similares a las descritas anteriormente para el vídeo 3D para el primer candidato entre capas y el segundo candidato entre capas.

25 **[0190]** En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, firmware o cualquier combinación de los mismos. Si se implementan en software, las funciones se pueden almacenar en, o transmitir por, un medio legible por ordenador, como una o más instrucciones o código, y ejecutarse mediante una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que correspondan a un medio tangible tal como medios de almacenamiento de datos, o medios de comunicación que incluyan cualquier medio que facilite la transferencia de un programa informático desde un lugar a otro, por ejemplo, de acuerdo a un protocolo de comunicación. De esta manera, los medios legibles por ordenador pueden corresponder en general a (1) medios de almacenamiento tangibles legibles por ordenador que sean no transitorios o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser medios disponibles cualesquiera a los que se pueda acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en esta divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

35 **[0191]** A modo de ejemplo, y no de limitación, tales medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento de disco óptico, almacenamiento de disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que pueda usarse para almacenar código de programa deseado en forma de instrucciones o estructuras de datos y al que pueda accederse mediante un ordenador. Además, cualquier conexión recibe adecuadamente la denominación de medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde una sede de la Red, un servidor u otro origen remoto usando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas se incluyen en la definición de medio. Sin embargo, debería entenderse que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que, en cambio, se orientan a medios de almacenamiento tangibles no transitorios. El término disco, como se usa en el presente documento, incluye el disco compacto (CD), el disco láser, el disco óptico, el disco versátil digital (DVD), el disco flexible y el disco Blu-ray, donde algunos discos habitualmente emiten datos magnéticamente, mientras que otros discos emiten datos ópticamente con láseres. Las combinaciones de lo anterior también se deberían incluir dentro del alcance de los medios legibles por ordenador.

60 **[0192]** 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 puertas programables in situ (FPGA) u otros circuitos lógicos, integrados o discretos,

5 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 decodificación, o incorporados en un códec combinado. Además, las técnicas se podrían implementar totalmente en uno o más circuitos o elementos lógicos.

10 **[0193]** Las técnicas de la presente divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluidos un equipo manual inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). Diversos componentes, módulos o unidades se describen en esta divulgación para enfatizar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no requieren necesariamente su realización mediante diferentes unidades de hardware. En cambio, como se ha descrito anteriormente, diversas unidades se pueden combinar en una unidad de hardware de códec, o ser proporcionadas por un grupo de unidades de hardware interoperativas, incluyendo uno o más procesadores, como se ha descrito anteriormente, conjuntamente con software y/o firmware adecuados.

15 **[0194]** Se han descrito diversos ejemplos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.

**REIVINDICACIONES**

1. Un procedimiento de codificación de datos de vídeo, comprendiendo el procedimiento:
- 5 determinar que un bloque actual dentro de un fragmento, imagen o secuencia tiene predicción inter en el modo de fusión o predicción inter en el modo de salto;
- basado en el bloque actual con predicción inter en el modo de fusión o con predicción inter en el modo de salto:
- 10 identificar un bloque de referencia, para el bloque actual, en una visualización de referencia a la que hace referencia un vector de disparidad desplazado un primer valor de desviación;
- determinar si un IPMVC o un IDMVC se incluirán en una lista de candidatos;
- 15 en el que el IPMVC comprende información de movimiento para el bloque de referencia en una visualización de referencia a la que hace referencia el vector de disparidad, para el bloque actual, desplazado el primer valor de desviación,
- 20 en el que el IDMVC comprende el vector de disparidad, para el bloque actual, desplazado un segundo valor de desviación que es diferente al primer valor de desviación utilizado para desplazar el vector de disparidad para identificar el bloque de referencia en el que el segundo valor de desviación comprende un valor de desviación fijo para el desplazamiento del vector de disparidad horizontalmente que se fija para todas las unidades de predicción dentro del fragmento, imagen o secuencia; y
- 25 determinar un segundo IPMVC, en el que el segundo IPMVC comprende información de movimiento para un segundo bloque de referencia en la visualización de referencia a la que hace referencia el vector de disparidad sin desplazar el vector de disparidad; y
- 30 comparar información de movimiento del primer IPMVC e información de movimiento del segundo IPMVC;
- en el que determinar si el IPMVC o el IDMVC se incluirá en la lista de candidatos comprende determinar que el primer IPMVC no se incluirá en la lista de candidatos si la información de movimiento para el
- 35 segundo IPMVC es la misma que la información de movimiento del primer IPMVC, y determinar que el primer IPMVC se incluirá en la lista de candidatos si la información de movimiento para el segundo IPMVC es diferente a la información de movimiento para el primer IPMVC; y
- codificación mediante predicción inter del bloque actual en una imagen actual en una visualización actual, diferente de la visualización de referencia, basada en la lista de candidatos.
- 40
2. El procedimiento según la reivindicación 1, en el que la comparación comprende comparar el primer IPMVC solo con el segundo IPMVC.
- 45
3. El procedimiento según la reivindicación 1, en el que la información de movimiento del bloque de referencia comprende al menos uno de un vector de movimiento y uno o más índices de referencia para listas de imágenes de referencia.
- 50
4. El procedimiento de acuerdo con la reivindicación 1, que comprende además:
- recibir un índice en la lista de candidatos,
- en el que determinar que el bloque actual tiene predicción inter en el modo de fusión o predicción inter en el modo de salto comprende la descodificación de elementos sintácticos que indican que el bloque actual
- 55 tiene predicción inter en el modo de fusión o predicción inter en el modo de salto, y
- en el que la codificación entre predicciones comprende la descodificación mediante predicción inter del bloque actual basándose en la información identificada por el índice recibido en la lista de candidatos.
- 60
5. El procedimiento según la reivindicación 1, en el que la codificación mediante predicción inter comprende la codificación mediante predicción inter del bloque actual basándose en la información en la lista de candidatos, comprendiendo además el procedimiento:
- 65 los elementos sintácticos de señalización que indican que el bloque actual tiene predicción inter en el modo de fusión o tiene predicción inter en modo de salto; y

señalar un índice en la lista de candidatos que se utiliza para identificar la información en la lista de candidatos.

6. Un dispositivo para codificar datos de vídeo que comprende:

- 5 medios para determinar que un bloque actual dentro de un fragmento, imagen o secuencia tiene predicción inter en el modo de fusión o tiene predicción inter en el modo de salto;
- 10 medios para identificar, para el bloque actual, un bloque de referencia en una visualización de referencia a la que hace referencia un vector de disparidad desplazado un primer valor de desviación;
- 15 medios para determinar si un IPMVC o un IDMVC se incluirán en una lista de candidatos, en el que el IPMVC comprende información de movimiento para el bloque de referencia en una visualización de referencia a la que hace referencia el vector de disparidad, para el bloque actual, desplazado el primer valor de desviación y en el que el IDMVC comprende el vector de disparidad, para el bloque actual, desplazado un segundo valor de desviación que es diferente al primer valor de desviación usado para desplazar el vector de disparidad para identificar el bloque de referencia en el que el segundo valor de desviación comprende un valor de desviación fijo para desplazar el vector de disparidad horizontalmente que está fijo para todas las unidades de predicción dentro del fragmento, imagen o secuencia;
- 20 medios para determinar un segundo IPMVC, en el que el segundo IPMVC comprende información de movimiento para un segundo bloque de referencia en la visualización de referencia a la que hace referencia el vector de disparidad sin desplazar el vector de disparidad;
- 25 medios para comparar información de movimiento del primer IPMVC e información de movimiento del segundo IPMVC,
- 30 en el que determinar si el IPMVC o el IDMVC se incluirá en la lista de candidatos comprende determinar que el primer IPMVC no se incluirá en la lista de candidatos si la información de movimiento para el segundo IPMVC es la misma que la información de movimiento del primer IPMVC, y determinar que el primer IPMVC se incluirá en la lista de candidatos si la información de movimiento para el segundo IPMVC es diferente a la información de movimiento para el primer IPMVC; y
- 35 medios para la codificación mediante predicción inter del bloque actual en una imagen actual en una visualización actual, diferente de la visualización de referencia, basándose en la lista de candidatos.

7. El dispositivo de la reivindicación 6, en el que el bloque de referencia comprende un primer bloque de referencia, el bloque actual comprende un primer bloque, el vector de disparidad comprende un primer vector de disparidad y la visualización de referencia comprende una primera visualización de referencia, comprendiendo además el dispositivo:

- 40 medios para determinar un primer IPMVC para un segundo bloque que tiene predicción inter en el modo de fusión o tiene predicción inter en modo de salto, en el que el primer IPMVC comprende información de movimiento para un segundo bloque de referencia en una segunda visualización de referencia a la que hace referencia un segundo vector de disparidad, para el segundo bloque, desplazado un tercer valor de desviación;
- 45 medios para determinar un segundo IPMVC para el segundo bloque, en el que el segundo IPMVC comprende información de movimiento para un tercer bloque de referencia en la segunda visualización de referencia a la que hace referencia el segundo vector de disparidad, para el segundo bloque, sin desplazar el segundo vector de disparidad;
- 50 medios para comparar el primer IPMVC solo con el segundo IPMVC;
- 55 medios para incluir el primer IPMVC en una lista de candidatos para el segundo bloque basándose en la comparación que indica que el segundo IPMVC es diferente al primer IPMVC; y
- 60 medios para la codificación mediante predicción inter del segundo bloque basándose en la lista de candidatos para el segundo bloque.

8. Un dispositivo para codificar datos de vídeo, comprendiendo el dispositivo un codificador de vídeo que comprende el dispositivo de la reivindicación 6.

9. El dispositivo de la reivindicación 8, en el que el dispositivo comprende uno de:

- 65 un circuito integrado (IC);

un microprocesador; y

un dispositivo de comunicación inalámbrica.

- 5
- 10.** Un medio de almacenamiento legible por ordenador que tiene instrucciones almacenadas en el mismo que, al ejecutarse, hacen que uno o más procesadores de un dispositivo para codificar datos de vídeo emprendan el procedimiento según cualquiera de las reivindicaciones 1 a 5.

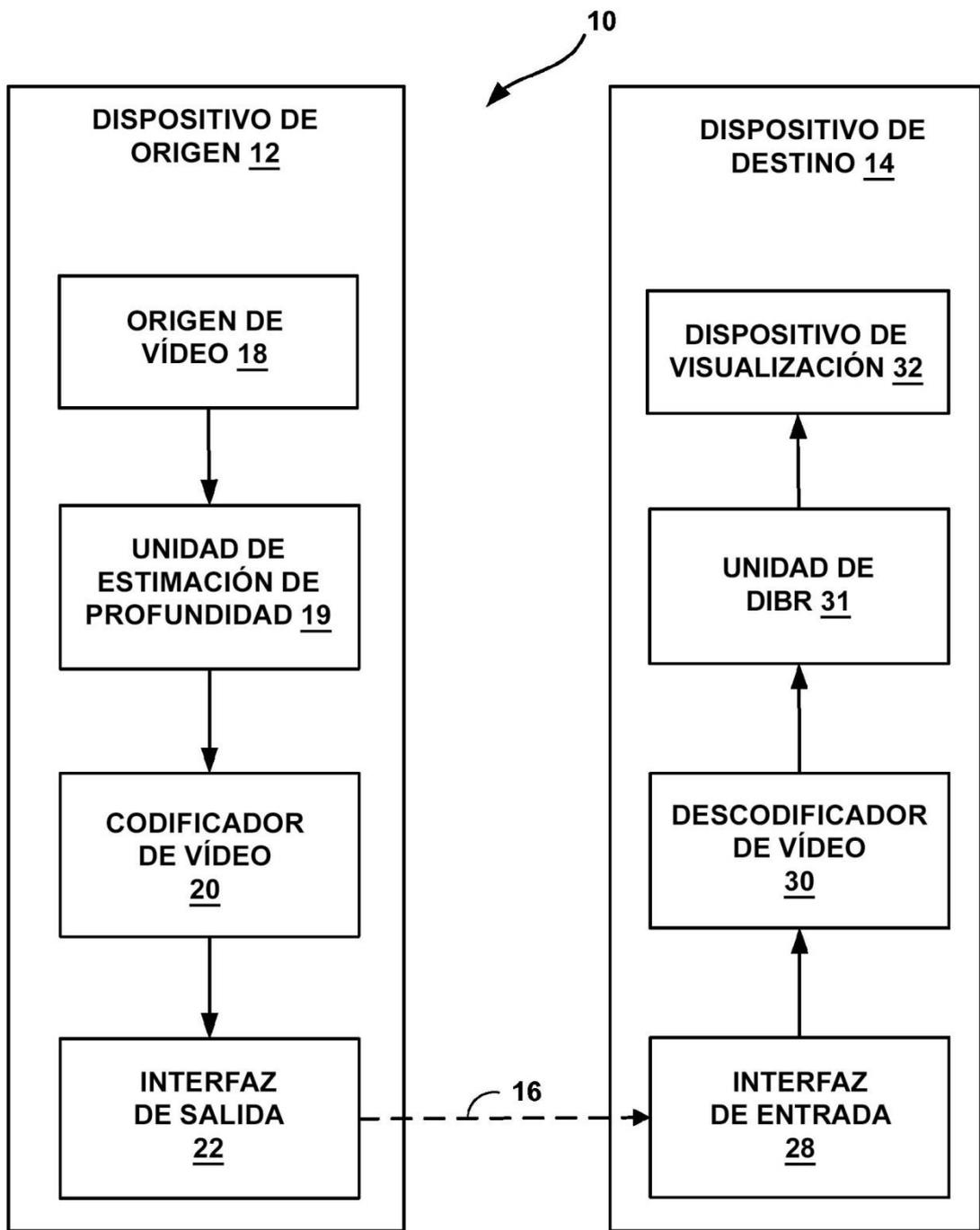
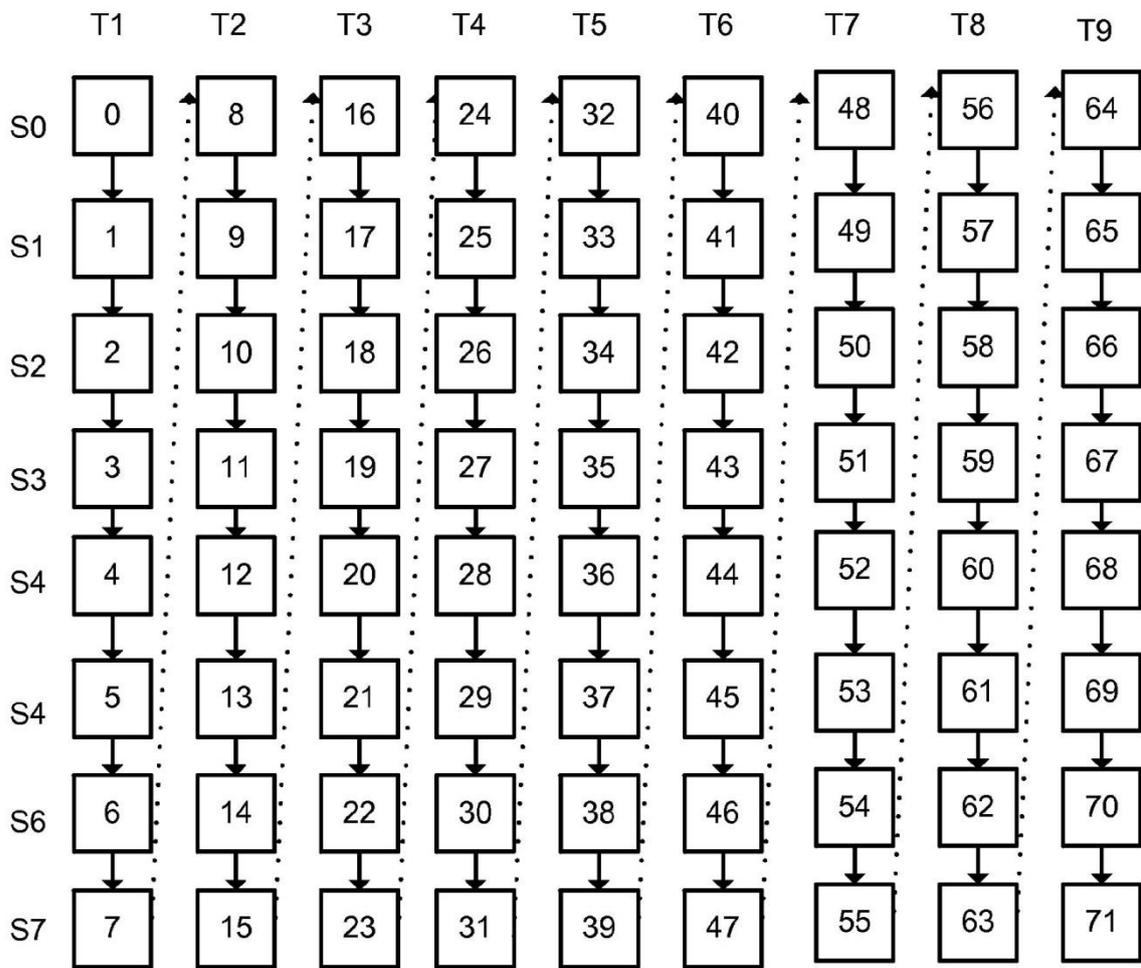


FIG. 1



**FIG. 2**

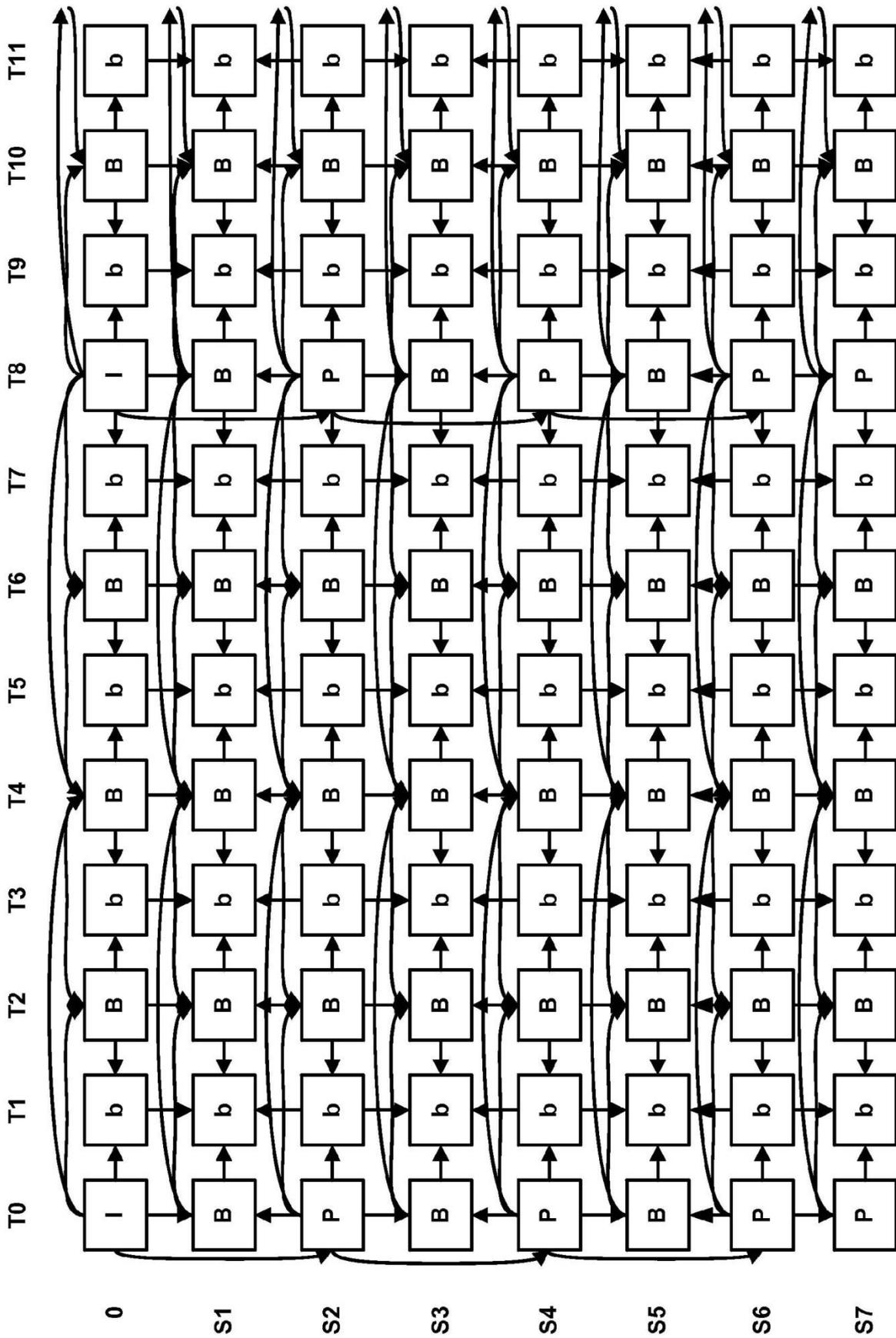


FIG. 3

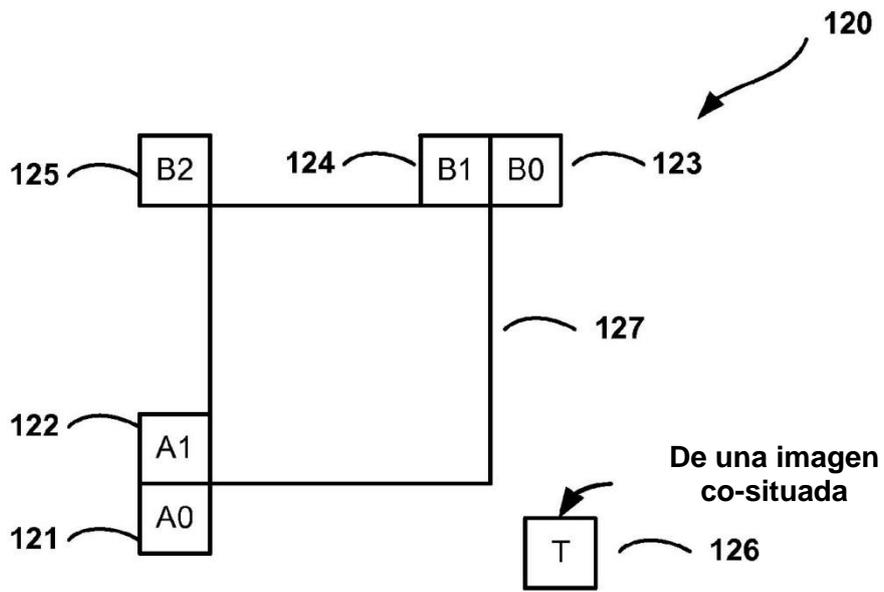


FIG. 4

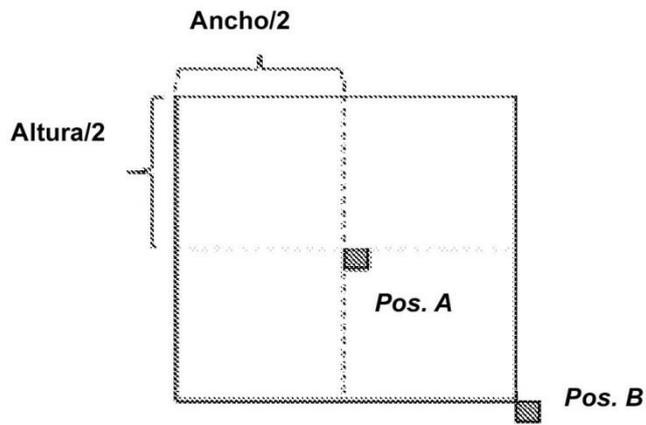


FIG. 5

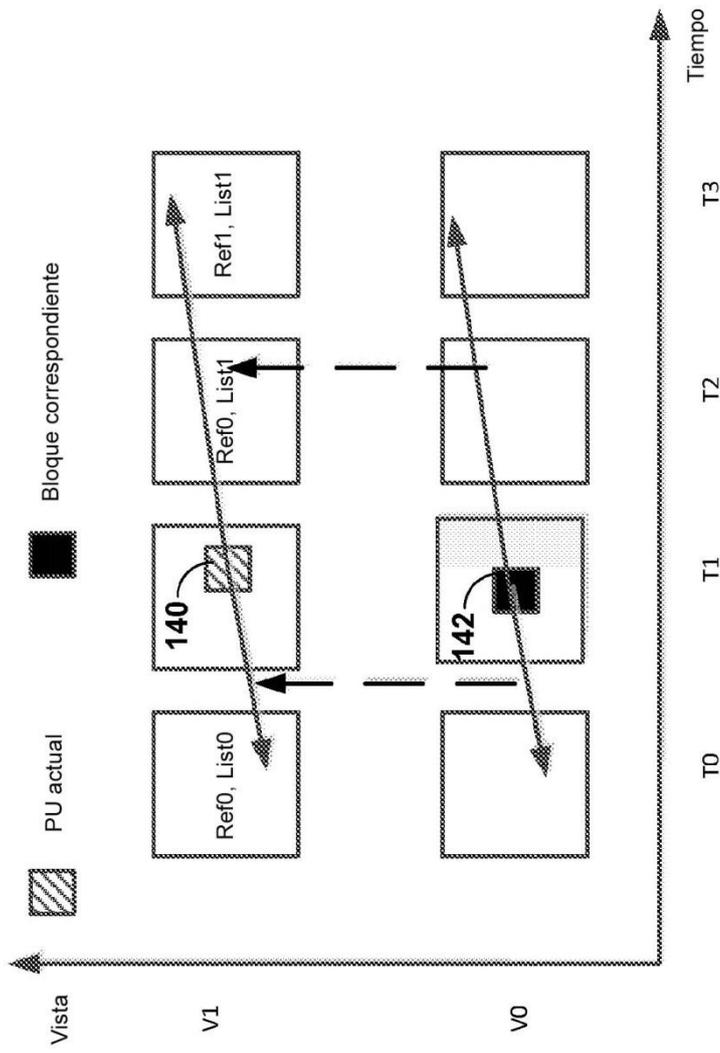


FIG. 6

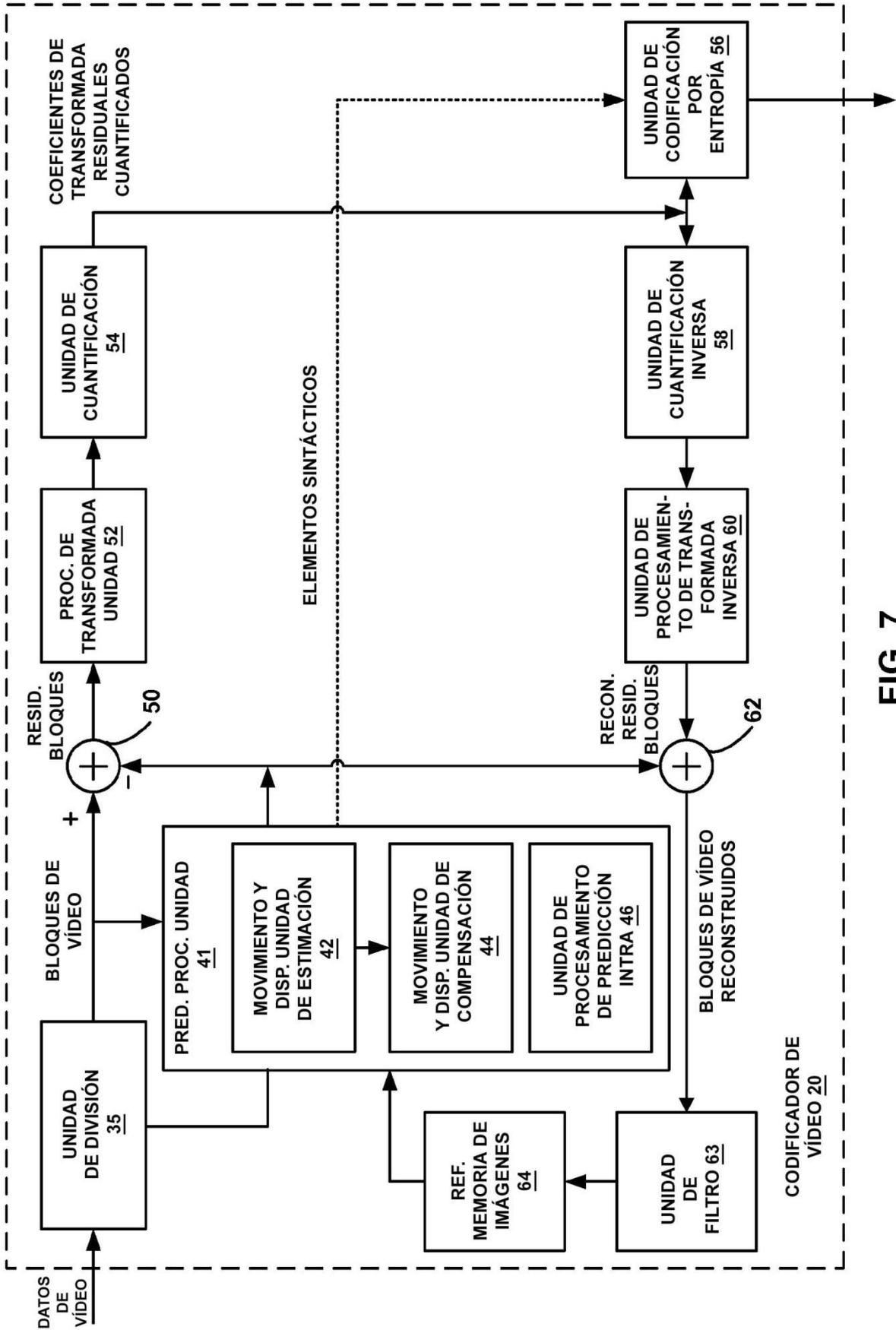


FIG. 7

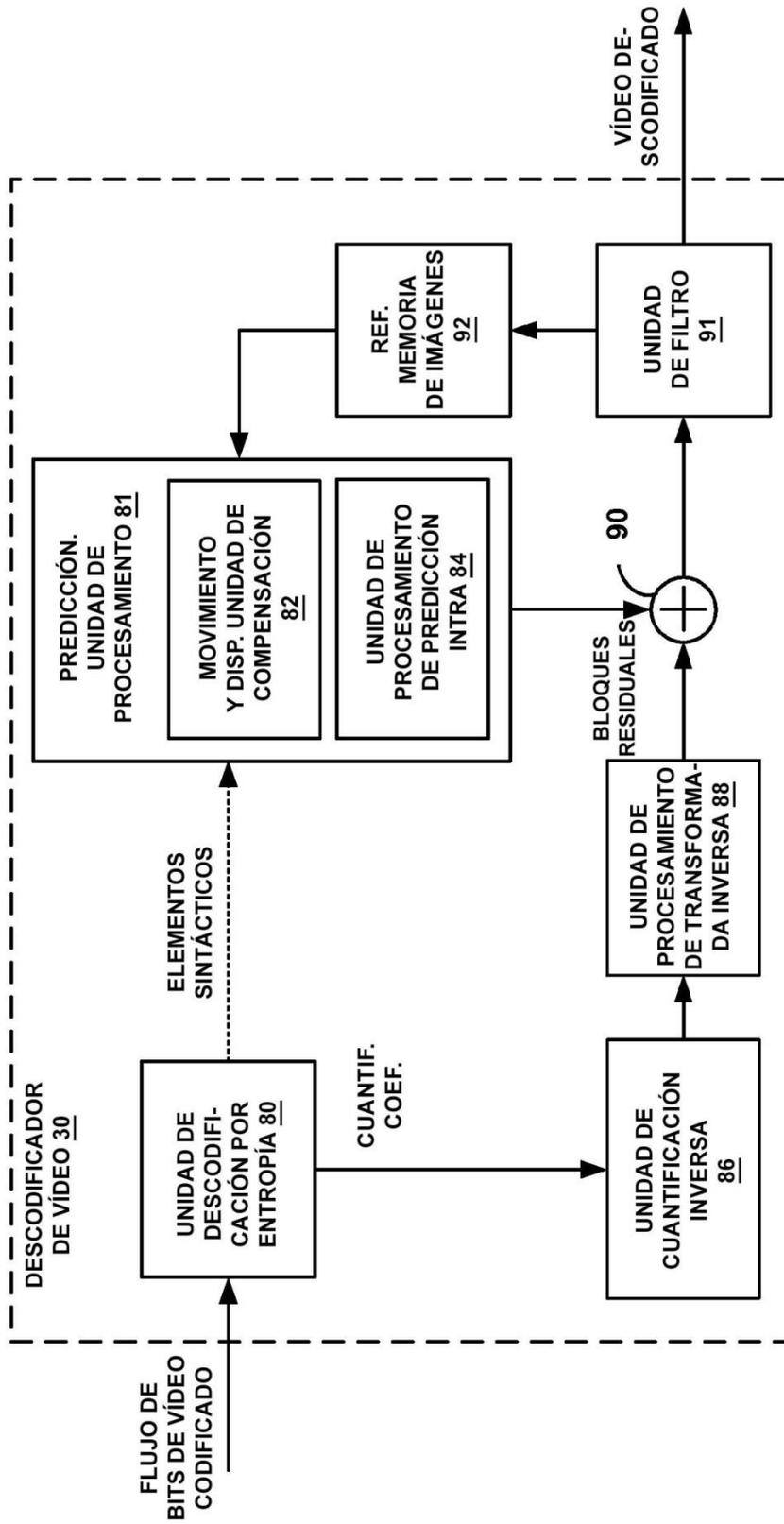


FIG. 8

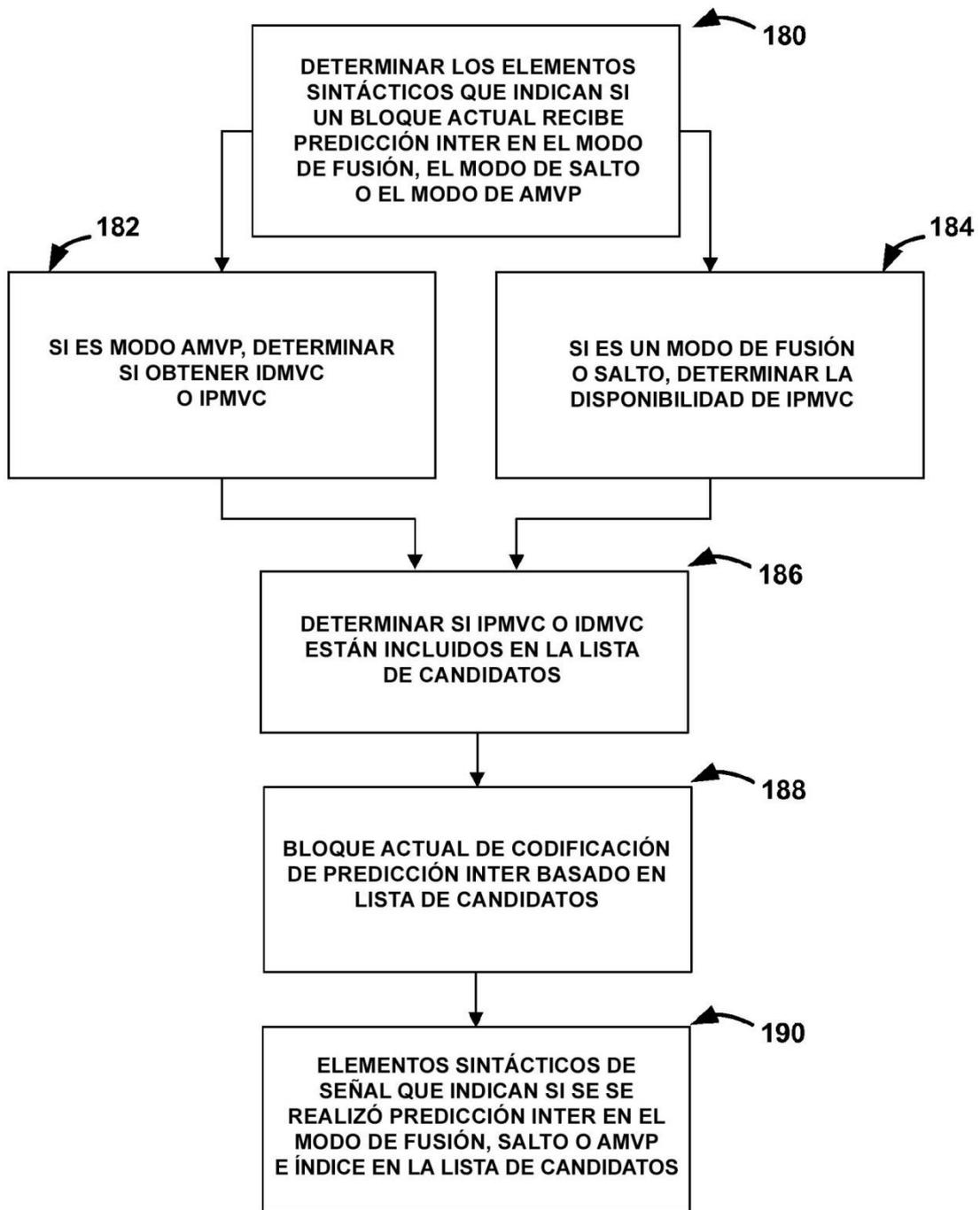


FIG. 9

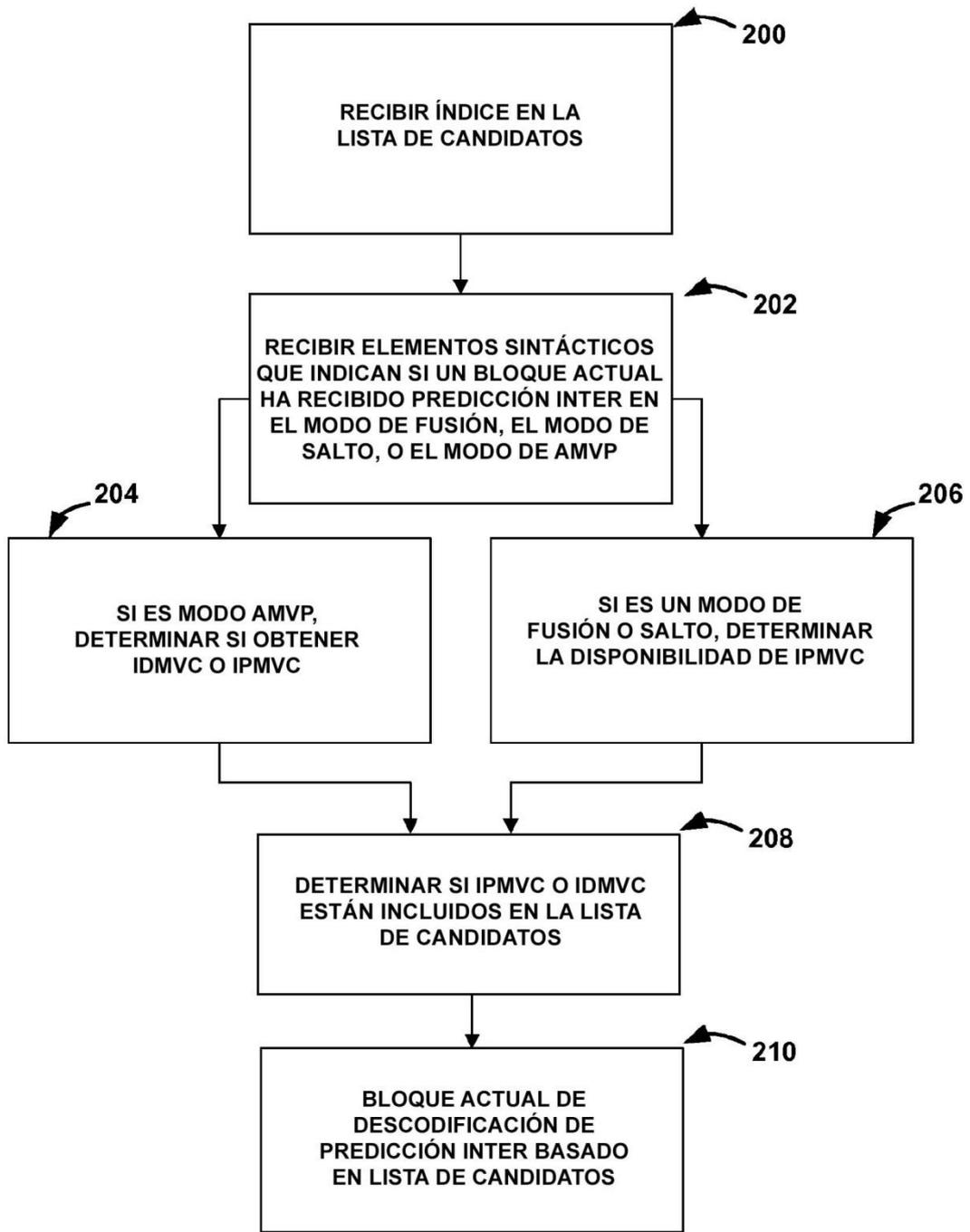


FIG. 10