

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 638 416**

51 Int. Cl.:

H04N 19/105 (2014.01)

H04N 19/176 (2014.01)

H04N 19/70 (2014.01)

H04N 19/513 (2014.01)

H04N 19/597 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **20.06.2014** **PCT/US2014/043403**

87 Fecha y número de publicación internacional: **24.12.2014** **WO14205343**

96 Fecha de presentación y número de la solicitud europea: **20.06.2014** **E 14740047 (7)**

97 Fecha y número de publicación de la concesión europea: **07.06.2017** **EP 3011746**

54 Título: **Predicción residual avanzada (ARP) más precisa para la codificación de texturas**

30 Prioridad:

21.06.2013 US 201361838208 P

14.07.2013 US 201361846036 P

19.06.2014 US 201414309711

45 Fecha de publicación y mención en BOPI de la
traducción de la patente:
20.10.2017

73 Titular/es:

QUALCOMM INCORPORATED (100.0%)

5775 Morehouse Drive

San Diego, CA 92121-1714, US

72 Inventor/es:

ZHANG, LI;

CHEN, YING y

KARCZEWICZ, MARTA

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 638 416 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

Predicción residual avanzada (ARP) más precisa para la codificación de texturas

- 5 Esta solicitud reivindica el beneficio de la solicitud provisional de Estados Unidos 61/838 208, presentada el 21 de junio de 2013, y de la solicitud provisional de Estados Unidos 61/846 036, presentada el 14 de julio de 2013.

CAMPO TÉCNICO

- 10 Esta divulgación se refiere a la codificación de vídeo.

ANTECEDENTES

- 15 Las capacidades del vídeo digital pueden incorporarse en una amplia gama de dispositivos, incluyendo 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, consolas de videojuegos, teléfonos celulares o de radio por satélite, los denominados "teléfonos inteligentes", dispositivos de videoconferencia, dispositivos de transmisión de vídeo y similares. Los dispositivos de vídeo digital implementan técnicas de codificación de vídeo, tales como las descritas en las normas definidas por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificación Avanzada de Vídeo (AVC), la norma de Codificación de Vídeo de Alta Eficiencia (HEVC), actualmente en desarrollo y las ampliaciones de tales normas. Los dispositivos de vídeo pueden transmitir, recibir, codificar, decodificar y/o almacenar información de vídeo digital más eficazmente, implementando tales técnicas de codificación de vídeo.

- 25 Las técnicas de codificación de vídeo incluyen la predicción espacial (intra-imagen) y/o la predicción temporal (entre imágenes) para reducir o eliminar la redundancia intrínseca en las secuencias de vídeo. Para la codificación de vídeo basándose 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 intracodificado (I) de una imagen son codificados usando la predicción espacial con respecto a muestras de referencia en bloques contiguos de la misma imagen. Los bloques de vídeo en un fragmento inter-codificado (P o B) de una imagen pueden usar la predicción espacial con respecto a muestras de referencia en bloques contiguos de 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.

- 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 a codificar y el bloque predictivo. Un bloque intercodificado se codifica de acuerdo con un vector de movimiento que apunta a un bloque de muestras de referencia que forman el bloque predictivo, y los datos residuales que indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intra-codificado se codifica de acuerdo con una modalidad de intra-codificación y los datos residuales. Para una mayor compresión, los datos residuales pueden transformarse desde el dominio de píxeles a un dominio de transformación, dando como resultado coeficientes de transformación residuales, los cuales pueden cuantificarse posteriormente. Los coeficientes de transformación cuantificados, inicialmente dispuestos en una formación bidimensional, pueden escanearse con el fin de producir un vector unidimensional de coeficientes de transformación, y puede aplicarse la codificación por entropía para lograr aún más compresión.

RESUMEN

- 50 En general, esta divulgación describe técnicas exactas de predicción residual avanzada (ARP) para la codificación de textura, que pueden proporcionar una precisión mejorada con relación a otras técnicas de ARP. Más particularmente, esta divulgación describe técnicas ARP que incluyen la identificación de un DMV desde la vista actual hasta una vista de referencia y la determinación de un bloque de predicción residual para ARP del bloque de vídeo actual basándose en la identificación del DMV.

- 55 En algunos ejemplos, el DMV es un DMV del bloque de vídeo actual, y las técnicas incluyen determinar un bloque de predicción residual de inter-vista para ARP de inter-vista del bloque de vídeo actual. El DMV se utiliza para la predicción de inter-vista del bloque de vídeo actual basándose en un bloque de vídeo de referencia de inter-vista. Las técnicas para ARP de inter-vista también pueden incluir la identificación de bloques de vídeo de referencia temporales en las vistas actuales y de referencia basándose en un vector de movimiento temporal (TMV) del bloque de vídeo de referencia de inter-vista, y la determinación de un bloque de predicción residual basándose en una diferencia entre los bloques de vídeo de referencia temporales. En dichos ejemplos, ARP no se limita a ARP temporal para codificar bloques de vídeo predichos temporalmente, y en su lugar puede incluir ARP de inter-vista para codificar bloques de vídeo predichos de inter-vista.

- 65 En algunos ejemplos, el bloque de vídeo actual se predice temporalmente y un DMV del bloque de vídeo de

referencia para el bloque de vídeo actual reemplaza a un vector de disparidad derivado para el bloque de vídeo actual, por ejemplo, de acuerdo con la obtención de vector de disparidad basándose en bloques contiguos (NBDV), para ARP temporal del bloque de vídeo actual. En tales ejemplos, el DMV, que se elige típicamente mediante optimización de la distorsión de velocidad, puede ser más preciso que el vector de disparidad derivado, lo cual
5 puede originar una ARP temporal más precisa del bloque de vídeo actual. En algunos ejemplos, el bloque de vídeo actual se predice temporalmente y un vector de disparidad obtenido a través de un bloque de profundidad co-ubicado del bloque de vídeo de referencia temporal para el bloque de vídeo actual reemplaza a un vector de disparidad derivado para el bloque de vídeo actual, por ejemplo de acuerdo con la Obtención de Vector de Disparidad basándose en Bloques Contiguos (NBDV), para ARP temporal del bloque de vídeo actual. Tales
10 ejemplos pueden proporcionar una ARP temporal más precisa cuando el bloque de profundidad co-ubicado está disponible durante la codificación de textura.

En un ejemplo, un procedimiento de predicción residual avanzada de inter-vista para descodificar datos de vídeo comprende descodificar un flujo de bits de vídeo codificado que codifica los datos de vídeo para identificar un vector
15 de movimiento de disparidad (DMV) y un bloque residual para un bloque de vídeo actual. El bloque de vídeo actual está en una vista actual y el DMV se utiliza para la predicción de inter-vista del bloque de vídeo actual basándose en un bloque de vídeo de referencia de inter-vista en una vista de referencia y en la misma unidad de acceso que el bloque de vídeo actual. El procedimiento comprende además identificar un vector de movimiento temporal (TMV) y una imagen de referencia asociada del bloque de vídeo de referencia de inter-vista, identificando un bloque de vídeo
20 de referencia temporal en la imagen de referencia asociada en la vista de referencia basándose en el TMV del bloque de vídeo de referencia de inter-vista, e identificar un bloque de vídeo de referencia temporal en la vista actual basándose en el TMV del bloque de vídeo de referencia de inter-vista en la vista de referencia. El bloque de vídeo de referencia temporal en la vista actual y el bloque de vídeo de referencia temporal en la vista de referencia se encuentran en la misma unidad de acceso. El procedimiento comprende además determinar un bloque de predicción residual para el bloque de vídeo actual basándose en una diferencia entre el bloque de vídeo de referencia temporal en la vista actual y el bloque de vídeo de referencia temporal en la vista de referencia. El procedimiento comprende
25 además aplicar el bloque de predicción residual y el bloque residual identificados desde el flujo de bits de vídeo codificado al bloque de vídeo de referencia de inter-vista para reconstruir el bloque de vídeo actual.

En otro ejemplo, un procedimiento de predicción residual avanzada de inter-vista para codificar datos de vídeo comprende identificar un vector de movimiento de disparidad (DMV) para un bloque de vídeo actual, en el que el
30 bloque de vídeo actual está en una vista actual y en el que el DMV se utiliza para predicción de inter-vista del bloque de vídeo actual basándose en un bloque de vídeo de referencia de inter-vista en una vista de referencia y la misma unidad de acceso que el bloque de vídeo actual. El procedimiento comprende además identificar un vector de movimiento temporal (TMV) y una imagen de referencia asociada del bloque de vídeo de referencia de inter-vista, identificando un bloque de vídeo de referencia temporal en la imagen de referencia asociada en la vista de
35 referencia basándose en el TMV del bloque de vídeo de referencia de inter-vista, e identificar un bloque de vídeo de referencia temporal en la vista actual basándose en el TMV del bloque de vídeo de referencia de inter-vista en la vista de referencia. El bloque de vídeo de referencia temporal en la vista actual y el bloque de vídeo de referencia temporal en la vista de referencia se encuentran en la misma unidad de acceso. El procedimiento comprende además determinar un bloque de predicción residual para el bloque de vídeo actual basándose en una diferencia entre el bloque de vídeo de referencia temporal en la vista actual y el bloque de vídeo de referencia temporal en la vista de referencia. El procedimiento comprende además codificar un flujo de bits de vídeo codificado que codifica
40 los datos de vídeo para identificar el DMV y un bloque residual para el bloque de vídeo actual. El bloque residual identificado por el flujo de bits de vídeo codificado comprende una diferencia entre el bloque de vídeo de referencia de inter-vista y el bloque de predicción residual para el bloque de vídeo actual.

En otro ejemplo, un aparato comprende un codificador de vídeo configurado para realizar una predicción residual avanzada de inter-vista para codificar datos de vídeo. El codificador de vídeo comprende una memoria configurada
50 para almacenar un flujo de bits de vídeo codificado que codifica los datos de vídeo y uno o más procesadores. El uno o más procesadores están configurados para identificar un vector de movimiento de disparidad (DMV) para un bloque de vídeo actual, en el que el bloque de vídeo actual está en una vista actual y en el que el DMV se utiliza para la predicción de inter-vista del bloque de vídeo actual basándose en un bloque de vídeo de referencia de inter-vista en una vista de referencia y en la misma unidad de acceso que el bloque de vídeo actual. El uno o más
55 procesadores están configurados adicionalmente para identificar un vector de movimiento temporal (TMV) y una imagen de referencia asociada del bloque de vídeo de referencia de inter-vista, identificar un bloque de vídeo de referencia temporal en la imagen de referencia asociada en la vista de referencia basándose en el TMV del bloque de vídeo de referencia de inter-vista e identificar un bloque de vídeo de referencia temporal en la vista actual basándose en el TMV del bloque de vídeo de referencia de inter-vista en la vista de referencia. El bloque de vídeo de referencia temporal en la vista actual y el bloque de vídeo de referencia temporal en la vista de referencia están situados en la misma unidad de acceso. El uno o más procesadores están configurados además para determinar un bloque de predicción residual para el bloque de vídeo actual basándose en una diferencia entre el bloque de vídeo de referencia temporal en la vista actual y el bloque de vídeo de referencia temporal en la vista de referencia. El uno
60 o más procesadores están configurados además para codificar el flujo de bits de vídeo codificado para identificar el DMV y un bloque residual para el bloque de vídeo actual. El bloque residual identificado mediante la codificación del flujo de bits de vídeo codificado comprende una diferencia entre el bloque de vídeo de referencia de inter-vista y el

bloque de predicción residual para el bloque de vídeo actual.

En otro ejemplo, un medio de almacenamiento legible por ordenador tiene instrucciones almacenadas en el mismo que, cuando se ejecutan, hacen que uno o más procesadores de un codificador de vídeo identifiquen un vector de movimiento de disparidad (DMV) para un bloque de vídeo actual, en el que el bloque de vídeo actual está en una vista actual, y en el que el DMV se utiliza para la predicción de inter-vista del bloque de vídeo actual basándose en un bloque de vídeo de referencia de inter-vista en una vista de referencia y en la misma unidad de acceso que el bloque de vídeo actual. Las instrucciones además hacen que uno o más procesadores identifiquen un vector de movimiento temporal (TMV) y una imagen de referencia asociada del bloque de vídeo de referencia de inter-vista, identifiquen un bloque de vídeo de referencia temporal en la imagen de referencia asociada en la vista de referencia basándose en el TMV del bloque de vídeo de referencia de inter-vista, e identifiquen un bloque de vídeo de referencia temporal en la vista actual basándose en el TMV del bloque de vídeo de referencia de inter-vista en la vista de referencia. El bloque de vídeo de referencia temporal en la vista actual y el bloque de vídeo de referencia temporal en la vista de referencia están situados en la misma unidad de acceso. Las instrucciones además hacen que uno o más procesadores determinen un bloque de predicción residual para el bloque de vídeo actual basándose en una diferencia entre el bloque de vídeo de referencia temporal en la vista actual y el bloque de vídeo de referencia temporal en la vista de referencia. Las instrucciones además hacen que uno o más procesadores codifiquen el flujo de bits de vídeo codificado para identificar el DMV y un bloque residual para el bloque de vídeo actual. El bloque residual identificado mediante la codificación del flujo de bits de vídeo codificado comprende una diferencia entre el bloque de vídeo de referencia de inter-vista y el bloque de predicción residual para el bloque de vídeo actual.

En otros ejemplos, los procedimientos de predicción residual avanzada de inter-vista para codificar datos de vídeo comprenden la identificación de un vector de movimiento de disparidad (DMV) para un bloque de vídeo actual, en el que el bloque de vídeo actual está en una vista actual y en el que el DMV se utiliza para predicción de inter-vista del bloque de vídeo actual basándose en un bloque de vídeo de referencia de inter-vista en una vista de referencia y la misma unidad de acceso que el bloque de vídeo actual. Los procedimientos comprenden además la identificación de un vector de movimiento temporal (TMV) y una imagen de referencia asociada. En algunos ejemplos, el DMV puede ser de una primera lista de imágenes de referencia del bloque de vídeo actual y el TMV y la imagen de referencia asociada pueden ser de una segunda lista de imágenes de referencia del bloque de vídeo actual. En otros ejemplos, el TMV y la imagen de referencia asociada se obtienen a partir de bloques contiguos espaciales o temporales del bloque de vídeo actual. En cualquier caso, los procedimientos pueden comprender además la identificación de un bloque de vídeo de referencia temporal en la vista de referencia basándose en el TMV, e identificar un bloque de vídeo de referencia temporal en la vista actual basándose en el TMV. Los procedimientos comprenden además la determinación de un bloque de predicción residual para el bloque de vídeo actual basándose en una diferencia entre el bloque de vídeo de referencia temporal en la vista actual y el bloque de vídeo de referencia temporal en la vista de referencia. Los procedimientos comprenden además codificar un flujo de bits de vídeo codificado que codifica los datos de vídeo para identificar el DMV y un bloque residual para el bloque de vídeo actual. El bloque residual identificado por el flujo de bits de vídeo codificado comprende una diferencia entre el bloque de vídeo de referencia de inter-vista y el bloque de predicción residual para el bloque de vídeo actual.

En otro ejemplo, un procedimiento de predicción residual temporal avanzada para codificar datos de vídeo comprende identificar un vector de movimiento temporal (TMV) para un bloque de vídeo actual, en el que el bloque de vídeo actual está en una vista actual y en el que el TMV se utiliza para la predicción del bloque de vídeo actual basándose en un bloque de vídeo de referencia temporal en la vista actual y en una unidad de acceso diferente al bloque de vídeo actual. El procedimiento comprende además identificar un vector de movimiento de disparidad (DMV) del bloque de vídeo de referencia temporal usado para la predicción de inter-vista del bloque de vídeo de referencia temporal. El procedimiento comprende además determinar, basándose en el DMV, por lo menos uno de un bloque de vídeo de referencia de inter-vista en la vista de referencia y en la misma unidad de acceso que el bloque de vídeo actual o un bloque de vídeo de referencia temporal en la vista de referencia y en la unidad de acceso diferente. El procedimiento comprende además determinar el bloque de predicción residual para el bloque de vídeo actual basándose en una diferencia entre el bloque de vídeo de referencia de inter-vista en la vista de referencia y en la misma unidad de acceso que el bloque de vídeo actual y el bloque de vídeo de referencia temporal en la vista de referencia y en la unidad de acceso diferente. El procedimiento comprende además codificar un flujo de bits de vídeo codificado que codifica los datos de vídeo para identificar el TMV y un bloque residual para el bloque de vídeo actual. El bloque residual identificado por el flujo de bits de vídeo codificado comprende una diferencia entre el bloque de vídeo de referencia temporal y el bloque de predicción residual para el bloque de vídeo actual. Algunos ejemplos de este procedimiento comprenden además escalar el TMV del bloque de vídeo actual a una imagen de referencia objetivo en una unidad de acceso objetivo para la predicción residual avanzada del bloque de vídeo actual, en el que el TMV escalado identifica el bloque de vídeo de referencia temporal en la vista actual. En algunos ejemplos de este procedimiento, el bloque de vídeo de referencia temporal en la vista actual identificado por el TMV escalado comprende un primer bloque de vídeo de referencia temporal y el procedimiento comprende además determinar que el primer bloque de vídeo de referencia temporal en la vista actual identificado por el TMV escalado no está asociado con un DMV, e identificar un segundo bloque de vídeo de referencia temporal en la vista actual basándose en el escalado ausente de TMV. En tales ejemplos, la identificación del DMV comprende identificar un DMV del segundo bloque de vídeo de referencia temporal en la vista actual identificado por el escalado

ausente de TMV. En algunos ejemplos de este procedimiento, el bloque de vídeo de referencia temporal en la vista actual y en una unidad de acceso diferente al bloque de vídeo actual comprende una pluralidad de unidades de predicción, e identificar el DMV del bloque de vídeo de referencia temporal comprende identificar un DMV asociado con una de la pluralidad de PU que contiene una posición central del bloque de vídeo de referencia temporal. En algunos ejemplos de este procedimiento, identificar el DMV comprende identificar un DMV de un modo de predicción distinto a la predicción de síntesis de vídeo inverso (BVSP). En algunos ejemplos de este procedimiento, el bloque de vídeo de referencia de inter-vista contiene un primer conjunto de información de movimiento correspondiente a una primera lista de imágenes de referencia y un segundo conjunto de información de movimiento correspondiente a una segunda lista de imágenes de referencia, ya la identificación del TMV del bloque de vídeo de referencia de inter-vista comprende seleccionar del TMV del primer conjunto de información de movimiento si el primer conjunto de información de movimiento incluye un TMV y seleccionar el TMV del segundo conjunto de información de movimiento si el primer conjunto de información de movimiento no incluye un TMV. En algunos ejemplos de este procedimiento, la primera lista de imágenes de referencia comprende RefPicList0. En algunos ejemplos de este procedimiento, un orden para considerar el primer y segundo conjuntos de información de movimiento es independiente de cuál de los primero y segundo conjuntos de información de movimiento incluye el TMV. En algunos ejemplos de este procedimiento, la codificación del flujo de bits de vídeo codificado comprende descodificar el flujo de bits de vídeo codificado con un descodificador de vídeo para identificar el TMV y el bloque residual para el bloque de vídeo actual y aplicar el bloque de predicción residual y el bloque residual identificado del flujo de bits de vídeo codificado al bloque de vídeo de referencia temporal para reconstruir el bloque de vídeo actual. En algunos ejemplos de este procedimiento, codificar el flujo de bits de vídeo codificado comprende codificar el flujo de bits de vídeo codificado con un codificador de vídeo para indicar el TMV y el bloque residual para el bloque de vídeo actual a un descodificador de vídeo.

En otro ejemplo, un procedimiento de predicción residual temporal avanzada para codificar datos de vídeo comprende identificar un vector de movimiento temporal (TMV) para un bloque de vídeo actual, en el que el bloque de vídeo actual está en una vista actual y en el que el TMV se utiliza para la predicción del bloque de vídeo actual basándose en un bloque de vídeo de referencia temporal en la vista actual y en una unidad de acceso diferente al bloque de vídeo actual. El procedimiento comprende además obtener un vector de disparidad (DV) a través de un bloque de profundidad co-ubicado del bloque de vídeo de referencia temporal. El procedimiento comprende además determinar, basándose en el DV, al menos uno de un bloque de vídeo de referencia de inter-vista en la vista de referencia y en la misma unidad de acceso que el bloque de vídeo actual o un bloque de vídeo de referencia temporal en la vista de referencia y en la unidad de acceso diferente. El procedimiento comprende además determinar un bloque de predicción residual para el bloque de vídeo actual basándose en una diferencia entre el bloque de vídeo de referencia de inter-vista en la vista de referencia y en la misma unidad de acceso que el bloque de vídeo actual y el bloque de vídeo de referencia temporal en la vista de referencia y en la unidad de acceso diferente. El procedimiento comprende además codificar un flujo de bits de vídeo codificado que codifica los datos de vídeo para identificar el TMV y un bloque residual para el bloque de vídeo actual. El bloque residual identificado por el flujo de bits de vídeo codificado comprende una diferencia entre el bloque de vídeo de referencia temporal y el bloque de predicción residual para el bloque de vídeo actual. Algunos ejemplos de este procedimiento comprenden además escalar el TMV del bloque de vídeo actual a una imagen de referencia objetivo en una unidad de acceso objetivo para la predicción residual avanzada del bloque de vídeo actual, en el que el TMV escalado identifica el bloque de vídeo de referencia temporal en la vista actual. En algunos ejemplos de este procedimiento, el bloque de vídeo de referencia temporal en la vista actual identificado por el TMV escalado comprende un primer bloque de vídeo de referencia temporal y el procedimiento comprende además determinar que el primer bloque de vídeo de referencia temporal en la vista actual identificado por el TMV escalado no está asociado con un DMV, e identificar un segundo bloque de vídeo de referencia temporal en la vista actual basándose en el escalado ausente de TMV. En tales ejemplos, la identificación del DMV comprende identificar un DMV del segundo bloque de vídeo de referencia temporal en la vista actual identificado por el escalado ausente de TMV. En algunos ejemplos de este procedimiento, obtener el DV comprende convertir un valor de profundidad de una muestra dentro del bloque de profundidad co-ubicado en el DV. En algunos ejemplos, la muestra se encuentra en $(W/2, H/2)$ con respecto a la muestra superior izquierda del bloque de profundidad co-ubicado, en el que el tamaño del bloque de profundidad co-ubicado es $W \times H$. En algunos ejemplos de este procedimiento, obtener el DV comprende determinar un valor de profundidad representativo basándose en valores de profundidad de una pluralidad de muestras dentro del bloque de profundidad co-ubicado y convertir el valor de profundidad representativo en el DV. En algunos ejemplos, la pluralidad de muestras son cuatro muestras de esquina. En algunos ejemplos, la pluralidad de muestras se selecciona basándose en muestras contiguas del bloque de profundidad. En algunos ejemplos, la determinación del valor de profundidad representativo basándose en valores de profundidad de una pluralidad de muestras dentro del bloque de profundidad co-ubicado comprende determinar el valor de profundidad representativo basándose en todos los valores de profundidad de una pluralidad de muestras dentro del bloque de profundidad co-ubicado. En algunos ejemplos de este procedimiento, la codificación del flujo de bits de vídeo codificado comprende descodificar el flujo de bits de vídeo codificado con un descodificador de vídeo para identificar el TMV y el bloque residual para el bloque de vídeo actual y aplicar el bloque de predicción residual y el bloque residual identificado del flujo de bits de vídeo codificado al bloque de vídeo de referencia temporal para reconstruir el bloque de vídeo actual. En algunos ejemplos de este procedimiento, codificar el flujo de bits de vídeo codificado comprende codificar el flujo de bits de vídeo codificado con un codificador de vídeo para indicar el TMV y el bloque residual para el bloque de vídeo actual a un descodificador de vídeo.

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

- 5 **BREVE DESCRIPCIÓN DE LOS DIBUJOS**
- 10 La FIG. 1 es un diagrama de bloques que ilustra un ejemplo de sistema de codificación y decodificación de vídeo que puede utilizar las técnicas descritas en esta divulgación.
- La FIG. 2 es un diagrama gráfico que ilustra un ejemplo de orden de codificación o decodificación multivista.
- 15 La FIG. 3 es un diagrama conceptual que ilustra un ejemplo de patrón de predicción temporal y de inter-vista para la codificación de vídeo multivista.
- La FIG. 4 es un diagrama conceptual que ilustra un ejemplo de relación de bloques contiguos con un bloque actual para la predicción de la información de movimiento del bloque actual.
- 20 La FIG. 5 es un diagrama conceptual que ilustra un ejemplo de obtención de un candidato de vector de movimiento predicho de inter-vista y un candidato de vector de movimiento de disparidad de inter-vista para la predicción de la información de movimiento del bloque actual.
- 25 La FIG. 6 es un diagrama conceptual que ilustra ejemplos de bloques contiguos espaciales, a partir de los cuales se puede obtener un vector de disparidad para un bloque de vídeo actual utilizando la Obtención de Vector de Disparidad Basándose en Bloques Contiguos (NBDV), en relación con el bloque de vídeo actual.
- 30 La FIG. 7 es un diagrama conceptual que ilustra un ejemplo de localización de un bloque de profundidad a partir de una vista de referencia y el uso del bloque de profundidad localizado en la vista de referencia para la Predicción de Síntesis de Vista Inversa (BVSP).
- La FIG. 8 es un diagrama conceptual que ilustra un ejemplo de estructura de predicción para la Predicción Residual Avanzada (ARP) temporal de un bloque de vídeo predicho temporalmente.
- 35 La FIG. 9 es un diagrama conceptual que ilustra un ejemplo de estructura de predicción bidireccional para ARP temporal.
- La FIG. 10 es un diagrama conceptual que ilustra un ejemplo de estructura de predicción para ARP de inter-vista de un bloque de vídeo predicho de inter-vista de acuerdo con las técnicas descritas en esta divulgación.
- 40 La FIG. 11 es un diagrama conceptual que ilustra un ejemplo de estructura de predicción para ARP temporal de un bloque de vídeo predicho temporalmente usando un vector de movimiento de disparidad (DMV) de acuerdo con las técnicas descritas en esta divulgación.
- 45 La FIG. 12 es un diagrama conceptual que ilustra técnicas de ejemplo para la identificación de un vector de movimiento temporal (TMV) o DMV en o adyacente a un bloque de vídeo de acuerdo con las técnicas descritas en esta divulgación.
- 50 Las FIGs. 13A a 13D son diagramas conceptuales que ilustran ejemplos de órdenes de escaneado para identificar TMV o DMV de acuerdo con las técnicas de esta divulgación.
- La FIG. 14 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo que puede implementar las técnicas descritas en esta divulgación.
- 55 La FIG. 15 es un diagrama de bloques que ilustra un ejemplo de decodificador de vídeo que puede utilizar las técnicas descritas en esta divulgación.
- La FIG. 16 es un diagrama de flujo que ilustra un ejemplo de procedimiento de ARP para decodificar un bloque de vídeo de acuerdo con las técnicas descritas en esta divulgación.
- 60 La FIG. 17 es un diagrama de flujo que ilustra un ejemplo de un procedimiento de ARP de inter-vista para decodificar un bloque de vídeo predicho de inter-vista de acuerdo con las técnicas descritas en esta divulgación.
- La FIG. 18 es un diagrama de flujo que ilustra un ejemplo de procedimiento de ARP temporal para decodificar un bloque de vídeo predicho temporalmente de acuerdo con las técnicas descritas en esta divulgación.
- 65 La FIG. 19 es un diagrama de flujo que ilustra un ejemplo de procedimiento de ARP para codificar un bloque de

vídeo de acuerdo con las técnicas descritas en esta divulgación.

La FIG. 20 es un diagrama de flujo que ilustra un ejemplo de procedimiento de ARP de inter-vista para codificar un bloque de vídeo predicho de inter-vista de acuerdo con las técnicas descritas en esta divulgación.

La FIG. 21 es un diagrama de flujo que ilustra un ejemplo de procedimiento de ARP temporal para codificar un bloque de vídeo predicho temporalmente de acuerdo con las técnicas descritas en esta divulgación.

La FIG. 22 es un diagrama de flujo que ilustra un procedimiento de ejemplo para identificar un DMV para ARP temporal de acuerdo con las técnicas descritas en esta divulgación.

La FIG. 23 es un diagrama de flujo que ilustra un procedimiento de ejemplo para identificar un TMV o DMV para ARP de acuerdo con las técnicas descritas en esta divulgación.

DESCRIPCIÓN DETALLADA

En general, esta divulgación está relacionada con la codificación de vídeo multivista, en la que los datos de vídeo codificados incluyen dos o más vistas. En algunos ejemplos, la codificación de vídeo multivista incluye un proceso de codificación de vídeo multivista más profundidad. En algunos ejemplos, la codificación multivista puede incluir codificación de vídeo tridimensional o 3D, y puede denominarse codificación de vídeo en 3D. Algunas técnicas divulgadas pueden aplicarse también a la codificación de vídeo distinta de la codificación de vídeo multivista o 3D, por ejemplo, codificación de vídeo escalable o codificación de vídeo de acuerdo con una memoria descriptiva de base de una norma de codificación de vídeo, por ejemplo, en el que los datos de vídeo no incluyen múltiples vistas o capas.

Esta divulgación también se refiere a la predicción de la señal residual de un bloque de vídeo, por ejemplo, Predicción Residual Avanzada (ARP). Más particularmente, esta divulgación describe técnicas para ARP más precisa de los componentes de textura de datos de vídeo multivista en una vista no de base. Las técnicas para ARP más precisa pueden incluir la identificación de un vector de movimiento de disparidad (DMV) desde la vista actual de un bloque de vídeo actual a una vista de referencia. El DMV es un vector de movimiento utilizado para la predicción de inter-vista de datos de vídeo en la vista actual, por ejemplo, del bloque de vídeo actual o un bloque de vídeo de referencia, datos de vídeo basados en la vista de referencia. Las técnicas pueden incluir además el uso del DMV identificado para identificar bloques de vídeo de referencia para ARP y determinar un bloque de predicción residual para el bloque de vídeo actual basándose en los bloques de vídeo de referencia identificados. Un bloque residual codificado para el bloque actual identificado en un flujo de bits de vídeo codificado puede ser una diferencia entre un bloque residual normal, que es la diferencia entre el bloque actual y un bloque de vídeo de referencia para el bloque de vídeo actual, y el bloque de predicción residual después del potencial escalado basándose en el índice de factor de ponderación codificado. En esta divulgación, el término "actual" se utiliza en general para identificar una vista, imagen o bloque que se está codificando actualmente. De este modo, un bloque actual en general representa un bloque de datos de vídeo codificados, en oposición a un bloque de vídeo ya codificado o en oposición a un bloque de vídeo no codificado todavía.

En algunos ejemplos, el DMV puede ser un DMV del bloque de vídeo actual, en cuyo caso el codificador de vídeo puede usar el DMV para identificar un bloque de referencia en la vista de referencia. En dichos ejemplos, las técnicas pueden incluir la determinación de un bloque de predicción residual de inter-vista para ARP de inter-vista del bloque de vídeo actual basándose en el DMV identificado. En dichos ejemplos, la ARP no se limita a ARP temporal para codificar bloques de vídeo predichos temporalmente, y en su lugar puede incluir ARP de inter-vista para codificar bloques de vídeo predichos de inter-vista. La ARP de inter-vista puede permitir que el codificador de vídeo calcule con mayor precisión un predictor residual de inter-vista en una unidad de acceso diferente para predecir el valor residual del bloque de vídeo actual.

En algunos ejemplos, el bloque de vídeo actual se predice temporalmente y el DMV puede ser un DMV de un bloque de referencia temporal en la misma vista que el bloque de vídeo actual. En dichos ejemplos, el codificador de vídeo puede utilizar el DMV en lugar de un vector de disparidad (DV) obtenido para el bloque de vídeo actual para identificar uno o ambos de un bloque de referencia de inter-vista del bloque de vídeo actual en la vista de referencia, o un bloque de referencia temporal en la vista de referencia para ARP temporal del bloque de vídeo actual. El codificador de vídeo puede usar el bloque o bloques identificados basados en el DMV para calcular con mayor exactitud un predictor residual temporal (calculado en una vista de referencia) para predecir el valor residual del bloque de vídeo actual. En tales ejemplos, el DMV, que se elige típicamente mediante optimización de la distorsión de velocidad, puede ser más preciso que el vector de disparidad derivado, lo cual puede originar una ARP temporal más precisa del bloque de vídeo actual.

Entre las normas de codificación de vídeo se incluyen ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 o ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual e ITU-T H.264 (también conocida como ISO/IEC MPEG-4 AVC), incluyendo sus ampliaciones de codificación de vídeo ajustable a escala (SVC) y de codificación de vídeo multivista (MVC). El último borrador conjunto de MVC se describe en "Codificación de vídeo avanzada para servicios

audiovisuales genéricos", Recomendación ITU-T H.264, marzo de 2010.

Recientemente, ha finalizado el diseño de una nueva norma de codificación de vídeo, concretamente la Codificación de Vídeo de Alta Eficiencia (HEVC), por parte el Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) del Grupo de Expertos en Codificación de Vídeo (VCEG) de ITU-T y el Grupo de Expertos en Imágenes en Movimiento (MPEG) de ISO/IEC. El último borrador de la memoria descriptiva HEVC, denominado HEVC WD10 a continuación, está disponible en: http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v34.zip. La cita completa de HEVC WD10 es Bross et al., "High Efficiency vídeo Coding (HEVC), borrador de memoria descriptiva textual 10 (para FDIS y última llamada)," JCTVC-L1003_v34, Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) of ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 12.ª reunión: Ginebra, Suiza, 14-23 de enero de 2013.

El JCT-3V también está desarrollando la extensión multivista para HEVC, a saber MV-HEVC. Un reciente Borrador de Trabajo (WD) de MV-HEVC, denominado MV-HEVC WD3 de aquí en adelante, está disponible en: http://phenix.itsudparis.eu/jct2/doc_end_user/documents/3_Geneva/wg11/JCT3V-C1004-v4.zip. La cita completa de MV-HEVC WD3 es: Tech et al., "Borrador de Texto 3 de MV-HEVC (ISO/IEC 23008-2:201x/PDAM2)", JCT3V-C1004_d3, Equipo de Colaboración Conjunta sobre Desarrollo de Extensión de Codificación 3Dvídeo de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 3.ª Reunión: Ginebra, Suiza, 17-23 de enero de 2013.

La extensión escalable a HEVC, llamada SHVC, también está siendo desarrollada por el JCT-VC. Un borrador de trabajo (WD) reciente de SHVC, denominado SHVC WD1 de aquí en adelante, está disponible en: http://phenix.intevry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1008-v1.zip. La cita completa de SHVC WD1 es: Chen et al., "Borrador de Texto 1 de SHVC", JCTVC-L1008, Equipo de Colaboración Conjunta sobre Codificación de Vídeo (JCT-VC) de la ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 12.ª Reunión: Ginebra, Suiza, 14-23 de enero de 2013.

Actualmente, un Equipo de Colaboración Conjunta en Codificación de Vídeo en 3D (JCT-3C) de VCEG y MPEG está desarrollando una norma 3DV basándose en HEVC, para el cual parte de los esfuerzos de estandarización incluye la estandarización de MV-HEVC y otra parte de los esfuerzos de estandarización incluye la estandarización de la codificación de vídeo 3D (3DV) basándose en HEVC (3DHEVC). Para 3D-HEVC, se pueden incluir y soportar nuevas herramientas de codificación, incluyendo aquellas a nivel de unidad de codificación / unidad de predicción, tanto para las vistas de textura como de profundidad. Un modelo de prueba de software de referencia reciente para 3D-HEVC (3D-HTM-7.0) se puede descargar desde el siguiente enlace: https://heve.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-7.0/.

Una cita completa para una descripción de software de referencia reciente, así como el borrador de trabajo de 3D-HEVC es la siguiente: Tech et al., "3D-HEVC Test Model 4", JCT3V-D1005_spec_v1, Equipo de Colaboración Conjunta sobre Desarrollo de Extensión de Codificación de Vídeo en 3D de la ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 4.ª reunión: Incheon, Corea, 20-26 abr. 2013. Esta descripción de software de referencia y borrador de trabajo de 3D-HEVC se puede descargar desde el siguiente enlace: http://phenix.itsudparis.eu/jct2/doc_end_user/documents/4_Incheon/wg11/JCT3V-D1005-v1.zip.

Las técnicas descritas en esta divulgación pueden implementarse mediante un codificador de vídeo que funciona de acuerdo con, por ejemplo, las extensiones MV-HEVC o 3D-HEVC de HEVC, o la extensión MVC de H.264. Sin embargo, las técnicas descritas en esta divulgación no se limitan a esas normas, y pueden ampliarse a otras normas de codificación de vídeo descritas en el presente documento, u otras normas de codificación de vídeo no mencionadas en el presente documento, incluyendo normas que proporcionan predicción residual en codificación de vídeo.

La FIG. 1 es un diagrama de bloques que ilustra un ejemplo de sistema de codificación y decodificación de vídeo de acuerdo con uno o más ejemplos descritos en esta divulgación. Por ejemplo, el sistema 10 incluye el dispositivo de origen 12 y el dispositivo objetivo 14. El dispositivo de origen 12 y el dispositivo objetivo 14 están configurados para implementar las técnicas descritas en esta divulgación. En algunos ejemplos, el sistema 10 puede configurarse para soportar la codificación, transmisión, almacenamiento, decodificación y/o presentación de datos de vídeo codificados, tales como datos de vídeo codificados de acuerdo con la norma HEVC, por ejemplo, como se describe en WD10, y sus extensiones, tales como, por ejemplo, las extensiones descritas en MV-HEVC WD3, SHVC WD1, 3D-HEVC Test Model 4, o similares. Sin embargo, las técnicas descritas en esta divulgación pueden ser aplicables a otras normas de codificación de vídeo u otras extensiones.

Como se muestra en la FIG. 1, el sistema 10 incluye un dispositivo de origen 12 que genera datos de vídeo codificado, a decodificar en un momento posterior por parte de un dispositivo objetivo 14. El dispositivo de origen 12 y el dispositivo objetivo 14 pueden comprender cualquiera entre una amplia gama de dispositivos, incluyendo ordenadores de sobremesa, ordenadores plegables (es decir, portátiles), ordenadores de tableta, decodificadores, 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, un dispositivo de transmisión de vídeo o similares. En algunos casos, el dispositivo de origen 12 y el

dispositivo objetivo 14 pueden estar equipados para la comunicación inalámbrica.

El dispositivo objetivo 14 puede recibir los datos de vídeo codificados que se van a decodificar, mediante un enlace 16. El enlace 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 objetivo 14. En un ejemplo, el enlace 16 puede comprender un medio de comunicación para permitir al dispositivo de origen 12 transmitir datos de vídeo codificados directamente al dispositivo objetivo 14 en tiempo real. Los datos de vídeo codificados pueden ser modulados de acuerdo con una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitidos al dispositivo objetivo 14. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrico o cableado, tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión física. El medio de comunicación puede formar parte de una red basándose en paquetes, tal como una red de área local, una red de área extensa o una red global tal como Internet. El medio de comunicación puede incluir enrutadores, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo de origen 12 al dispositivo objetivo 14.

En algunos ejemplos, pueden emitirse datos codificados desde la interfaz de salida 22 hasta un dispositivo de almacenamiento 36. De forma similar, se puede acceder a los datos codificados desde el dispositivo de almacenamiento 34 mediante una interfaz de entrada 28. El dispositivo de almacenamiento 36 puede incluir cualquiera de entre una diversidad de medios de almacenamiento de datos de acceso distribuido o local, tales como una unidad de disco fijo, 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 codificado. En un ejemplo adicional, el dispositivo de almacenamiento 36 puede corresponder a un servidor de archivos o a otro dispositivo de almacenamiento intermedio que pueda retener el vídeo codificado generado por el dispositivo de origen 12. El dispositivo objetivo 14 puede acceder a los datos de vídeo almacenados desde el dispositivo de almacenamiento 36 a través de transmisión en continuo 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 objetivo 14. Entre los ejemplos de servidores de archivos se incluyen un servidor web (por ejemplo, para un sitio web), un servidor FTP, dispositivos de almacenamiento conectados a la red (NAS) o una unidad de disco local. El dispositivo objetivo 14 puede acceder a los datos de vídeo codificados a través de cualquier conexión de datos estándar, incluyendo una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión de Wi-Fi), una conexión por cable (por ejemplo, DSL, módem de cable, etc.), o una combinación de ambos que sea adecuada para acceder a datos de vídeo codificado, almacenados en un servidor de ficheros. La transmisión de datos de vídeo codificados desde el dispositivo de almacenamiento 36 puede ser una transmisión en continuo, una transmisión de descarga o una combinación de ambas.

Las técnicas de esta divulgación no están limitadas, por supuesto, a aplicaciones o configuraciones inalámbricas. Las técnicas pueden aplicarse a la codificación de vídeo, en soporte de cualquiera entre una diversidad de aplicaciones de multimedia, tales como difusiones de televisión por el aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo por flujo, por ejemplo, mediante Internet, codificación de vídeo digital para su almacenamiento en un medio de almacenamiento de datos, decodificación de vídeo digital almacenado en un medio de almacenamiento de datos, u otras aplicaciones. En algunos ejemplos, el sistema 10 puede configurarse para dar soporte a la transmisión de vídeo unidireccional o bidireccional, para prestar soporte a aplicaciones tales como la transmisión de vídeo, la reproducción de vídeo, la difusión de vídeo y/o la videotelefonía.

En el ejemplo de la FIG. 1, el dispositivo de origen 12 incluye una fuente de vídeo 18, un codificador de vídeo 20 y una interfaz de salida 22. En algunos casos, la interfaz de salida 22 puede incluir un modulador / desmodulador (módem) y/o un transmisor. En el dispositivo de origen 12, la fuente de vídeo 18 puede incluir una fuente tal como un dispositivo de captura de vídeo, por ejemplo, una videocámara, un archivo de vídeo que contiene vídeo previamente capturado, una interfaz de alimentación de vídeo para recibir vídeo desde un proveedor de contenido de vídeo y/o un sistema de gráficos de ordenador para generar datos de gráficos de ordenador como el vídeo de origen, o una combinación de tales orígenes. En un ejemplo, si la fuente de vídeo 18 es una videocámara, el dispositivo de origen 12 y el dispositivo objetivo 14 pueden formar los denominados teléfonos con cámara o videoteléfonos. Sin embargo, las técnicas descritas en esta divulgación pueden ser aplicables a la codificación de vídeo en general, y pueden aplicarse a aplicaciones inalámbricas y/o cableadas.

El vídeo capturado, pre-capturado o generado por ordenador puede ser codificado por el codificador de vídeo 12. Los datos de vídeo codificados pueden ser transmitidos directamente al dispositivo objetivo 14 mediante la interfaz de salida 22 del dispositivo de origen 12. Los datos de vídeo codificados también (o de forma alternativa) pueden almacenarse en el dispositivo de almacenamiento 36 para un acceso posterior por el dispositivo objetivo 14 u otros dispositivos, para decodificación y/o reproducción.

El dispositivo objetivo 14 incluye una interfaz de entrada 28, un decodificador de vídeo 30 y un dispositivo de visualización 32. En algunos casos, la interfaz de entrada 28 puede incluir un receptor y/o un módem. La interfaz de entrada 28 del dispositivo objetivo 14 recibe los datos de vídeo codificados por el enlace 16. Los datos de vídeo codificado, comunicados por el enlace 16, o proporcionados en el dispositivo de almacenamiento 36, pueden incluir una diversidad de elementos sintácticos generados por el codificador de vídeo 20, para su uso por un decodificador

de vídeo, tal como el decodificador de vídeo 30, en la decodificación de los datos de vídeo. Tales elementos sintácticos pueden incluirse con los datos de vídeo codificado, transmitidos en un medio de comunicación, almacenarse en un medio de almacenamiento o almacenarse en un servidor de ficheros.

- 5 El dispositivo de visualización 32 puede estar integrado con, o ser externo a, el dispositivo objetivo 14. En algunos ejemplos, el dispositivo objetivo 14 puede incluir un dispositivo de visualización integrado y también estar configurado para interconectarse con un dispositivo de visualización externo. En otros ejemplos, el dispositivo objetivo 14 puede ser un dispositivo de visualización. En general, el dispositivo de visualización 32 visualiza los datos de vídeo decodificado a un usuario, y puede comprender cualquiera entre una variedad de dispositivos de visualización, tales como 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.

- 10 El codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar de acuerdo con una norma de compresión de vídeo, tal como la norma HEVC actualmente en desarrollo, así como extensiones de la norma HEVC, por ejemplo, MV-HEVC, SHVC y 3D-HEVC. Sin embargo, las técnicas de esta divulgación no están limitadas a ninguna norma de codificación particular.

- 20 Aunque no se muestra en la FIG. 1, en algunos aspectos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden estar integrado, cada uno de ellos, con un codificador y decodificador de audio, y pueden incluir unidades adecuadas de multiplexado y de demultiplexado, u otro hardware y software, para gestionar la codificación, tanto de audio como de vídeo, en un flujo de datos común o en flujos de datos diferentes. Si procede, en algunos ejemplos, las unidades MUX-DEMUX pueden ajustarse al protocolo de multiplexado ITU H.223 o a otros protocolos, tales como el protocolo de datagramas de usuario (UDP).

- 25 El codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementarse como cualquiera entre una variedad de circuitos de codificadores adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados de aplicación específica (ASIC), matrices de puertas programables in situ (FPGA), lógica discreta, software, hardware, firmware o cualquier combinación de estos. Cuando las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en unos medios legibles por ordenador no transitorios adecuados, y ejecutar las instrucciones en hardware mediante uno o más procesadores que realizan las técnicas de esta divulgación. Tanto el codificador de vídeo 20 como el decodificador de vídeo 30 pueden estar incluidos en uno o más codificadores o decodificadores, donde cualquiera de ambos puede estar integrado como parte de un codificador/decodificador (CÓDEC) combinado en un dispositivo respectivo.

- 35 El codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar, en general, de acuerdo con HEVC WD10, MVHEVCWD3, SHVC WD1 y/o 3D-HEVC Test Model 4, como se ha descrito anteriormente, o con otras normas o extensiones similares en los que las técnicas descritas en esta divulgación pueden ser útiles. La norma HEVC especifica varias capacidades adicionales de los dispositivos de codificación de vídeo respecto a dispositivos existentes de acuerdo con, por ejemplo, la ITU-T H.264/AVC. Por ejemplo, mientras que la norma H.264 proporciona nueve modos de codificación de intra-predicción, la norma HEVC puede proporcionar hasta treinta y tres modos de codificación de intra-predicción.

- 45 En general, una trama o imagen de vídeo puede dividirse en una secuencia de bloques arbolados o unidades de codificación de mayor tamaño (LCU), que incluyen muestras tanto de luma como de croma. Un bloque arbolado en el proceso de codificación de HEVC tiene un fin similar al de un macrobloque de la norma H.264. Un fragmento incluye un número de bloques arbolados consecutivos en orden de codificación. Una trama o imagen de vídeo puede dividirse en uno o más fragmentos. Cada bloque arbolado puede separarse en unidades de codificación (CU) de acuerdo con un árbol cuádruple. Por ejemplo, un bloque arbolado, como un nodo raíz del árbol cuaternario, puede separarse en cuatro nodos hijo, y cada nodo hijo puede a su vez ser un nodo padre y dividirse en otros cuatro nodos hijo. Un nodo hijo final, no separado, como un nodo hoja del árbol cuaternario, comprende un nodo de codificación, es decir, un bloque de vídeo codificado. Los datos sintácticos asociados a un flujo de bits codificado pueden definir un número máximo de veces que puede separarse un bloque arbolado, y también pueden definir un tamaño mínimo de los nodos de codificación.

- 55 Una CU incluye un nodo de codificación y unidades de predicción (PU) y unidades de transformación (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 8 x 8 píxeles hasta el tamaño del bloque arbolado, con un máximo de 64 x 64 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 dependiendo de si la CU está codificada en modo de omisión o directo, codificada en modo de intra-predicción o codificada en modo de inter-predicción. Las PU pueden dividirse para no tener forma 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.

- 65 La norma HEVC admite transformaciones de acuerdo con las TU, que pueden ser diferentes para diferentes CU. El

tamaño de las TU típicamente se basa en el tamaño de las PU de una CU dada definida para una LCU dividida, aunque puede que no siempre sea así. Las TU presentan típicamente el mismo tamaño o un tamaño más pequeño que las PU. En algunos ejemplos, las muestras residuales correspondientes a una CU pueden subdividirse en unidades más pequeñas mediante una estructura de árbol cuádruple conocida como "árbol cuádruple residual" (RQT). Los nodos hoja del RQT pueden denominarse unidades de transformación (TU). Los valores de diferencias de píxeles asociados a las TU pueden transformarse para generar coeficientes de transformación, que pueden cuantificarse.

En general, una PU incluye datos relacionados con el proceso de predicción. Por ejemplo, cuando la PU está codificada en modalidad intra, la PU puede incluir datos que describen una modalidad de intra-predicción para la PU. En otro ejemplo, cuando la PU está codificada en la modalidad inter, la PU puede incluir datos que definen un vector de movimiento para la PU. Los datos que definen el vector de movimiento para una PU pueden describir, por ejemplo, un componente horizontal del vector de movimiento, un componente vertical del vector de movimiento, una resolución para el vector de movimiento (por ejemplo, precisión de píxeles de un cuarto o precisión de píxeles de un octavo), una imagen de referencia a la que apunta el vector de movimiento y/o una lista de imágenes de referencia (por ejemplo, RefPicList0 (L0) o RefPicList1 (L1)) para el vector de movimiento.

En general, se usa una TU para los procesos de transformación y cuantificación. Una CU dada que presenta una o más PU también puede incluir una o más unidades de transformación (TU). Tras la predicción, el codificador de vídeo puede calcular valores residuales correspondientes a la PU. Los valores residuales comprenden valores de diferencias de píxeles que se pueden transformar en coeficientes de transformación, cuantificar y escanear mediante las TU, para generar coeficientes de transformación en serie para la codificación por entropía. Esta divulgación usa típicamente el término "bloque de vídeo" para referirse a un nodo de codificación de una CU. En algunos casos específicos, esta divulgación también puede usar el término "bloque de vídeo" para referirse a un bloque arbolado, es decir, una LCU o una CU, que incluye un nodo de codificación y unas PU y TU.

Por ejemplo, para la codificación de vídeo de acuerdo con la norma HEVC, una trama de vídeo puede dividirse en unidades de codificación (CU), unidades de predicción (PU) y unidades de transformación (TU). Una CU en general se refiere a una región de la imagen que sirve como una unidad básica sobre la cual se aplican diversas herramientas de codificación para la compresión de vídeo. Una CU tiene típicamente una geometría cuadrada, y puede considerarse similar a un llamado "macro-bloque", en virtud de otras normas de codificación de vídeo, tales como, por ejemplo, la ITU-T H.264.

Para lograr una mejor eficiencia de codificación, una CU puede tener un tamaño variable dependiendo de los datos de vídeo que contiene. Es decir, una CU se puede partir, o "dividir" en bloques más pequeños, o sub-CU, cada uno de los cuales también se puede denominar CU. Además, cada CU que no está dividida en sub-CU puede dividirse en una o más PU y TU con fines de predicción y transformación de la CU, respectivamente.

Las PU pueden considerarse similares a las llamadas particiones de un bloque bajo otras normas de codificación de vídeo, tales como H.264. Las PU son la base sobre la cual se realiza la predicción para el bloque para producir coeficientes "residuales". Los coeficientes residuales de una CU representan una diferencia entre los datos de vídeo de la CU y los datos predichos para la CU determinados usando una o más PU de la CU. Específicamente, la una o más PU especifican cómo se divide la CU con el propósito de predicción y qué modo de predicción se utiliza para predecir los datos de vídeo contenidos dentro de cada partición de la CU.

Una o más TU de una CU especifican particiones de un bloque de coeficientes residuales de la CU basándose en las cuales se aplica una transformación al bloque para producir un bloque de coeficientes de transformación residual para la CU. La o las TU también pueden asociarse con el tipo de transformación que se aplica. La transformación convierte los coeficientes residuales de un dominio de píxel o espacial, a un dominio de transformación, como un dominio de frecuencia. Además, la una o más TU pueden especificar parámetros basándose en los cuales se aplica la cuantificación al bloque resultante de coeficientes de transformación residual para producir un bloque de coeficientes de transformación residual cuantificados. Los coeficientes de transformación residual pueden cuantificarse para posiblemente reducir la cantidad de datos utilizados para representar los coeficientes.

Una CU en general incluye un componente de luminancia, denominado Y, y dos componentes de crominancia, denominados U y V. En otras palabras, una CU dada que no está dividida en sub-CU puede incluir componentes Y, U y V, cada uno de los cuales se puede dividir adicionalmente en una o más PU y TU con fines de predicción y transformación de la CU, como se ha descrito anteriormente. Por ejemplo, dependiendo del formato de muestreo de vídeo, el tamaño de los componentes U y V, en términos del número de muestras, puede ser el mismo o diferente al tamaño del componente Y. Como tales, las técnicas descritas anteriormente con referencia a predicción, transformación y cuantificación pueden realizarse para cada uno de los componentes Y, U y V de una CU dada.

Para codificar una CU, se obtienen primero uno o más predictores para la CU basándose en una o más PU de la CU. Un predictor es un bloque de referencia que contiene datos predichos para la CU, y se obtiene basándose en una PU correspondiente para la CU, como se ha descrito anteriormente. Por ejemplo, la PU indica una partición de la CU para la que se tienen que determinar los datos predichos, y un modo de predicción utilizado para determinar

los datos predichos. Puede obtenerse el predictor, ya sea mediante los modos de intra- (I) predicción (es decir, predicción espacial) o la inter- (P o B) predicción (es decir, predicción temporal). Por lo tanto, algunas CU pueden estar intra-codificadas (I) usando la predicción espacial con respecto a los bloques de referencia contiguos, o CU, en la misma trama, mientras que otras CU pueden estar inter-codificadas (P o B) con respecto a los bloques de referencia; CU, en otras tramas.

Tras la identificación del uno o más predictores basados en una o más PU de la CU, se calcula una diferencia entre los datos de vídeo originales de la CU correspondientes a una o más PU y los datos predichos para la CU contenidos en uno o más predictores. Esta diferencia, también denominada valor residual de predicción, comprende coeficientes residuales y se refiere a diferencias de píxeles entre porciones de la CU especificadas por una o más PU y los uno o más predictores, tal como se ha descrito anteriormente. Los coeficientes residuales se disponen en general en una matriz bidimensional (2-D) que corresponde a una o más PU de la CU.

Para lograr una mejor compresión, el valor residual de predicción en general se transforma, por ejemplo, usando una transformación de coseno discreta (DCT), una transformación entera, una transformación de Karhunen-Loeve (KL) u otra transformación. La transformación convierte el valor residual de predicción, es decir, los coeficientes residuales, en el dominio espacial en coeficientes de transformación residual en el dominio de transformación, por ejemplo, un dominio de frecuencia, como también se ha descrito anteriormente. Los coeficientes de transformación también se disponen en general en una matriz 2-D que corresponde a la una o más TU de la CU. Para la compresión adicional, los coeficientes de transformación residual pueden cuantificarse para posiblemente reducir la cantidad de datos utilizados para representar los coeficientes, como también se ha descrito anteriormente.

Para lograr una compresión aún mayor, un codificador por entropía codifica posteriormente los coeficientes de transformación residual resultantes, utilizando Codificación Aritmética Binaria Adaptativa de Contexto (CABAC), Codificación de Longitud Variable Adaptativa de Contexto (CAVLC), codificación por entropía de Partición de Intervalo de Probabilidad (PIPE) u otra metodología de codificación por entropía. La codificación por entropía puede conseguir esta compresión adicional reduciendo o eliminando la redundancia estadística inherente en los datos de vídeo de la CU, representada por los coeficientes, con relación a otras CU.

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, en una cabecera de una o más de las imágenes o en otras ubicaciones, que describen un cierto número 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. Un codificador de vídeo 20 actúa típicamente sobre bloques de vídeo 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 de una CU. Los bloques de vídeo pueden presentar tamaños fijos o variables y pueden diferir en tamaño de acuerdo con una norma de codificación especificada.

En un ejemplo, la HEVC soporta la predicción en diversos tamaños de PU. Suponiendo que el tamaño de una CU específica sea $2N \times 2N$, la HEVC soporta la intra-predicción en tamaños de PU de $2N \times 2N$ o $N \times N$, y la inter-predicción en tamaños de PU simétrica de $2N \times 2N$, $2N \times N$, $N \times 2N$ o $N \times N$. LA HEVC también soporta la división asimétrica para la inter-predicción 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 25 % y 75 %. La parte de la CU correspondiente a la división de 25 % está indicada por una "n" seguida de una indicación "arriba", "abajo", "izquierda" o "derecha". Así, 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.

En esta divulgación, " $N \times N$ " y "N por N" pueden usarse de manera intercambiable para hacer referencia a las dimensiones de píxeles de un bloque de vídeo, en términos de dimensiones verticales y horizontales, por ejemplo, 16×16 píxeles o 16 por 16 píxeles. En general, un bloque de 16×16 tendrá 16 píxeles en una dirección vertical ($y = 16$) y 16 píxeles en la 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 en un bloque pueden estar ordenados en filas y columnas. Además, los bloques no necesitan presentar necesariamente el mismo número de píxeles en la dirección horizontal y en la dirección vertical. Por ejemplo, los bloques pueden comprender $N \times M$ píxeles, donde M no es necesariamente igual a N.

Tras la codificación de intra-predicción o inter-predicción mediante las PU de una CU, el codificador de vídeo 20 puede calcular datos residuales para las TU de la CU. Las PU pueden comprender datos de píxeles en el dominio espacial (también denominado dominio de píxeles) y las TU pueden comprender coeficientes en el dominio de transformación tras la aplicación de una transformación, por ejemplo, una transformación de coseno discreta (DCT), una transformación entera, una transformación de wavelet o una transformación similar desde un punto de vista conceptual 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 transformación para la CU.

Tras cualquier transformación para generar coeficientes de transformación, el codificador de vídeo 20 puede realizar la cuantificación de los coeficientes de transformación. La cuantificación se refiere en general a un proceso en el que los coeficientes de transformación se cuantifican para reducir posiblemente la cantidad de datos usados para representar los coeficientes, proporcionando una compresión adicional. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos o 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 .

En algunos ejemplos, el codificador de vídeo 20 puede usar un orden de escaneado predefinido para escanear los coeficientes de transformación cuantificados y generar un vector en serie que pueda someterse a la codificación por entropía. En otros ejemplos, el codificador de vídeo 20 puede realizar un escaneado adaptativo. Después de escanear los coeficientes de transformación cuantificados para formar un vector unidimensional, el codificador de vídeo 20 puede realizar la codificación por entropía del vector unidimensional, por ejemplo, de acuerdo con la codificación de longitud variable adaptativa de acuerdo con el contexto (CAVLC), la codificación aritmética binaria adaptativa según el contexto (CABAC), la codificación aritmética binaria adaptativa según el contexto basándose en la sintaxis (SBAC), la codificación por entropía por división de intervalos de probabilidad (PIPE) u otros procedimientos de codificación por entropía. El codificador de vídeo 20 también puede realizar la codificación por entropía de 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.

Para realizar la CABAC, el codificador de vídeo 20 puede asignar un contexto de un modelo contextual a un símbolo que se va a transmitir. El contexto puede referirse, 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 que se va a transmitir. Las palabras de código en la VLC pueden construirse de forma que los códigos relativamente más cortos correspondan a símbolos más probables, mientras que los códigos más largos correspondan a símbolos menos probables. De esta manera, el uso de la VLC puede permitir 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 puede basarse en un contexto asignado al símbolo.

Además, el codificador de vídeo 20 puede enviar datos sintácticos, tales como datos sintácticos basándose en bloques, datos sintácticos basándose en tramas y datos sintácticos basándose en GOP, al descodificador de vídeo 30, por ejemplo, en una cabecera de trama, una cabecera de bloque, una cabecera de fragmento o una cabecera de GOP. Los datos sintácticos de GOP pueden describir un número de tramas en el respectivo GOP, y los datos sintácticos de trama pueden indicar una modalidad de codificación/predicción utilizada para codificar la trama correspondiente.

Además, el codificador de vídeo 20 puede descodificar o reconstruir imágenes codificadas, por ejemplo, mediante cuantificación inversa y transformación inversa de datos residuales, y combinar los datos residuales con datos de predicción. De esta manera, el codificador de vídeo 20 puede simular el proceso de descodificación realizado por el descodificador de vídeo 30. Tanto el codificador de vídeo 20 como el descodificador de vídeo 30, por lo tanto, tendrán acceso a esencialmente las mismas imágenes descodificadas o reconstruidas, para su uso en la predicción entre imágenes.

En general, el descodificador de vídeo 30 puede realizar un proceso de descodificación que es el inverso al proceso de codificación realizado por el codificador de vídeo. Por ejemplo, el descodificador de vídeo 30 puede realizar la descodificación por entropía usando la inversa de las técnicas de codificación por entropía usadas por el codificador de vídeo para codificar por entropía los datos de vídeo cuantificados. El descodificador de vídeo 30 puede además cuantificar inversamente los datos de vídeo usando la inversa de las técnicas de cuantificación empleadas por el codificador de vídeo 20, y puede realizar una inversa de la transformación usada por el codificador de vídeo 20 para producir los coeficientes de transformación que están cuantificados. El descodificador de vídeo 30 puede entonces aplicar los bloques residuales resultantes a los bloques de referencia adyacentes (intra-predicción) o a bloques de referencia procedentes de otra imagen (inter-predicción), para producir el bloque de vídeo para su exhibición eventual. El descodificador de vídeo 30 puede ser configurado, instruido, controlado o dirigido para realizar la inversa de los diversos procesos realizados por el codificador de vídeo 20, basándose en los elementos sintácticos proporcionados por el codificador de vídeo 20, con los datos de vídeo codificados en el flujo de bits recibido por el descodificador de vídeo 30. Tal como se utiliza en el presente documento, el término "codificador de vídeo" puede referirse a un codificador de vídeo, tal como un codificador de vídeo 20, o un descodificador de vídeo, tal como un descodificador de vídeo 30. Además, los términos "codificación de vídeo" o "codificación" pueden referirse a cualquiera o ambas de codificación, por ejemplo, mediante un codificador de vídeo, o descodificación, por ejemplo, mediante un descodificador de vídeo.

En algunos ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 (FIG. 1) pueden emplear técnicas para la codificación de vídeo multivista, por ejemplo, la codificación de datos de vídeo que incluyen dos o más vistas. En tales ejemplos, el codificador de vídeo 20 puede codificar un flujo de bits que incluye datos de vídeo codificados para dos o más vistas, y el descodificador de vídeo 30 puede descodificar los datos de vídeo codificados para

proporcionar las dos o más vistas, por ejemplo, al dispositivo de visualización 32. En algunos ejemplos, el descodificador de vídeo 30 puede proporcionar las múltiples vistas de datos de vídeo para permitir al dispositivo de visualización 32 exhibir vídeo tridimensional. En algunos ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden ser conformes a una extensión 3D-HEVC de la norma HEVC, por ejemplo, en la cual se usan procesos de codificación multivista y de codificación multivista con profundidad. La codificación de vídeo multivista, o tridimensional, puede implicar la codificación de dos o más vistas de textura, y/o vistas que incluyan componentes de textura y de profundidad. En algunos ejemplos, los datos de vídeo codificados por el codificador de vídeo 20 y descodificados por el descodificador de vídeo 30 incluyen dos o más imágenes en cualquier instancia temporal dada, es decir, dentro de una “unidad de acceso”, o datos a partir de los cuales pueden obtenerse dos o más imágenes en cualquier instancia temporal dada.

En algunos ejemplos, un dispositivo, por ejemplo, la fuente de vídeo 18, puede generar las dos o más imágenes, por ejemplo, usando dos o más cámaras espacialmente desplazadas, u otros dispositivos de captura de vídeo, para capturar una escena común. Dos imágenes de la misma escena capturadas simultáneamente, o casi simultáneamente, desde posiciones horizontales levemente distintas, pueden ser usadas para producir un efecto tridimensional. En algunos ejemplos, la fuente de vídeo 18 (u otro componente del dispositivo de origen 12) puede usar información de profundidad o información de disparidad para generar una segunda imagen (u otra adicional) de una segunda vista (u otra adicional) en una instancia temporal dada, a partir de una primera imagen de una primera vista en la instancia temporal dada. En este caso, una vista dentro de una unidad de acceso puede incluir un componente de textura correspondiente a una primera vista y un componente de profundidad que puede ser usado, con el componente de textura, para generar una segunda vista. La información de profundidad o disparidad puede ser determinada por un dispositivo de captura de vídeo que captura la primera vista, por ejemplo, basándose en parámetros de cámara u otra información conocida, con respecto a la configuración del dispositivo de captura de vídeo y a la captura de los datos de vídeo para la primera vista. La información de profundidad o disparidad puede ser calculada, adicionalmente o de forma alternativa, por la fuente de vídeo 18 u otro componente del dispositivo de origen 12, a partir de parámetros de cámara y/o datos de vídeo en la primera vista.

Para presentar vídeo tridimensional, el dispositivo de visualización 32 puede exhibir, simultáneamente, o casi simultáneamente, dos imágenes asociadas a distintas vistas de una escena común, que fueron capturadas simultáneamente, o casi simultáneamente. En algunos ejemplos, un usuario del dispositivo objetivo 14 puede usar gafas activas para obturar, rápidamente y de forma alternativa, las lentes izquierda y derecha, y el dispositivo de visualización 32 puede conmutar rápidamente entre una vista izquierda y una vista derecha, en sincronización con las gafas activas. En otros ejemplos, el dispositivo de visualización 32 puede exhibir las dos vistas simultáneamente, y el usuario puede usar gafas pasivas, por ejemplo, con lentes polarizadas, que filtran las vistas para hacer que las vistas adecuadas pasen a través hasta los ojos del usuario. En otros ejemplos, el dispositivo de visualización 32 puede comprender un visor auto-estereoscópico, que no requiere gafas para que el usuario perciba el efecto tridimensional.

La codificación de vídeo multivista se refiere a la manera en la que se codifica una pluralidad de vistas. En el caso de la codificación de vídeo en 3D, la pluralidad de vistas puede, por ejemplo, corresponder a una vista de ojo izquierdo y una vista de ojo derecho. Cada vista de la pluralidad de vistas incluye una pluralidad de imágenes. La percepción del espectador de una escena 3D se debe a la disparidad horizontal entre los objetos en las imágenes de las diferentes vistas.

Un vector de disparidad (DV) para un bloque actual de una imagen actual es un vector que apunta a un bloque correspondiente en una imagen correspondiente que está en una vista diferente a la imagen actual. Por tanto, utilizando un DV, un codificador de vídeo puede localizar, en una imagen correspondiente, el bloque de aquél que corresponde a un bloque actual de una imagen actual. En este caso, la imagen correspondiente es una imagen que es de la misma instancia temporal que la imagen actual pero que está en una vista diferente. El bloque correspondiente en la imagen correspondiente y el bloque actual en la imagen actual pueden incluir contenido de vídeo similar; sin embargo, hay al menos una disparidad horizontal entre la ubicación del bloque actual en la imagen actual y la ubicación del bloque correspondiente en la imagen correspondiente. El DV del bloque actual proporciona una medida de esta disparidad horizontal entre el bloque en la imagen correspondiente y el bloque actual en la imagen actual.

En algunos casos, también puede haber disparidad vertical entre la ubicación del bloque dentro de la imagen correspondiente y la ubicación del bloque actual dentro de la imagen actual. El DV del bloque actual también puede proporcionar una medida de esta disparidad vertical entre el bloque en la imagen correspondiente y el bloque actual en la imagen actual. Un DV contiene dos componentes (un componente x y un componente y), aunque en muchos casos el componente vertical será igual a cero. El tiempo en que se muestra la imagen actual de la vista actual y la imagen correspondiente de la vista diferente puede ser el mismo, es decir, la imagen actual y las imágenes correspondientes son imágenes de la misma instancia temporal.

En la codificación de vídeo, en general hay dos tipos de predicción, comúnmente denominados intra-predicción e inter-predicción. En la intra-predicción, un codificador de vídeo predice un bloque de vídeo en una imagen basándose en un bloque ya codificado en la misma imagen. En la inter-predicción, un codificador de vídeo predice

un bloque de vídeo en una imagen basándose en un bloque ya codificado de una imagen diferente (es decir, una imagen de referencia). Una imagen de referencia, tal como se utiliza en esta divulgación, se refiere en general a cualquier imagen que contenga muestras que puedan ser utilizadas para la inter-predicción en el proceso de descodificación de imágenes posteriores en orden de descodificación. Cuando se codifica un contenido multivista, por ejemplo de acuerdo con 3D-HEVC, con respecto a una imagen actual, las imágenes de referencia pueden ser de la misma instancia temporal pero en una vista diferente o pueden estar en la misma vista pero de una instancia temporal diferente. En el caso de la codificación multivista, por ejemplo, en la 3D-HEVC, la predicción entre imágenes puede incluir la predicción del bloque de vídeo actual, por ejemplo, el nodo de codificación actual de una CU, a partir de otro bloque de vídeo en una imagen temporalmente distinta, es decir, a partir de una unidad de acceso distinta a la de la imagen actual, así como la predicción a partir de una imagen distinta en la misma unidad de acceso que la imagen actual, pero asociada a una vista distinta a la de la imagen actual.

En el último caso de la inter-predicción, puede denominarse codificación inter-vista o predicción de inter-vista. En la codificación multivista, la predicción de inter-vista se realiza entre las imágenes capturadas en las diferentes vistas de la misma unidad de acceso (es decir, con la misma instancia de tiempo) para eliminar la correlación entre las vistas. Al codificar una imagen de una vista no de base, por ejemplo, una vista dependiente, una imagen de la misma unidad de acceso, pero de una vista distinta, por ejemplo, de una vista de referencia, como la vista de base, puede añadirse a una lista de imágenes de referencia. Una imagen de referencia de inter-vista puede ponerse en cualquier posición de una lista de imágenes de referencia, como es el caso con cualquier imagen de referencia de inter-predicción (por ejemplo, temporal o inter-vista).

El bloque de la imagen de referencia utilizado para predecir el bloque de la imagen actual se identifica mediante un vector de movimiento. En la codificación multivista, hay al menos dos clases de vectores de movimiento. Un vector de movimiento temporal (TMV) es un vector de movimiento que apunta a un bloque en una imagen de referencia temporal que está en la misma vista que el bloque que está siendo codificado, pero en una instancia de tiempo o unidad de acceso diferente al bloque que está siendo codificado, y la inter-predicción correspondiente se denomina predicción compensada por movimiento (MCP). Otro tipo de vector de movimiento es un vector de movimiento de disparidad (DMV), que apunta a un bloque en una imagen en la misma imagen actual de unidad de acceso, pero de una vista distinta. Con un DMV, la correspondiente inter-predicción se denomina predicción compensada por disparidad (DCP) o predicción de inter-vista.

La FIG. 2 es un diagrama gráfico que ilustra un ejemplo de orden de codificación o descodificación multivista. La disposición de orden de descodificación ilustrada en la FIG. 2 puede denominarse codificación de primera vez. En general, una secuencia de vídeo en 3D o multivista puede incluir, para cada unidad de acceso (es decir, con la misma instancia de tiempo), dos o más imágenes para cada una de dos o más vistas, respectivamente. En la FIG. 2, S0-S7 se refieren a diferentes vistas del vídeo multivista. Cada T0-T8 representa una instancia de tiempo de salida. Una unidad de acceso puede incluir las imágenes codificadas de todas las vistas para una instancia de tiempo de salida. Por ejemplo, una primera unidad de acceso incluye todas las vistas S0-S7 para la instancia de tiempo T0 (es decir, las imágenes 0-7), una segunda unidad de acceso incluye todas las vistas S0-S7 para la instancia de tiempo T1 (es decir, las imágenes 8-15), etcétera. En este ejemplo, las imágenes 0-7 están en una misma instancia de tiempo (es decir, la instancia de tiempo T0), las imágenes 8-15 en una misma instancia de tiempo (es decir, la instancia de tiempo T1). Las imágenes con la misma instancia de tiempo se muestran en general al mismo tiempo, y es la disparidad horizontal, y posiblemente alguna disparidad vertical, entre los objetos dentro de las imágenes de la misma instancia de tiempo las que hacen que el espectador perciba una imagen que abarca un volumen en 3D.

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

El componente de vista de textura incluye el contenido de imagen real que se muestra. Por ejemplo, el componente de vista de textura puede incluir los componentes de luma (Y) y croma (Cb y Cr). El componente de vista de profundidad puede indicar profundidades relativas de los píxeles en su componente de vista de textura correspondiente. Como un ejemplo, el componente de vista de profundidad puede ser similar a una imagen en escala de grises que incluye únicamente valores de luma. En otras palabras, el componente de vista de profundidad tal vez no transmita ningún contenido de imagen, pero en lugar de eso proporcione una medida de las profundidades relativas de los píxeles en el componente de vista de textura.

Por ejemplo, un valor de píxel correspondiente a un píxel puramente blanco en el componente de vista de

profundidad puede indicar que su correspondiente píxel o píxeles en el componente de vista de textura correspondiente está más cerca de la perspectiva del espectador, y un valor de píxel correspondiente a un píxel puramente negro en el componente de vista de profundidad puede indicar que su correspondiente píxel o píxeles en el componente de vista de textura correspondiente está más alejado de la perspectiva del espectador. Los valores de píxel correspondientes a los diversos tonos de gris entre negro y blanco indican diferentes niveles de profundidad. Por ejemplo, un píxel muy gris en el componente de vista de profundidad indica que su píxel correspondiente en el componente de vista de textura está más alejado que un píxel ligeramente gris en el componente de vista de profundidad. Dado que solo se necesita un valor de píxel, similar a la escala de grises, para identificar la profundidad de píxeles, el componente de vista de profundidad puede incluir solo un valor de píxel. Por lo tanto, no son necesarios valores análogos a los componentes de croma.

El componente de vista de profundidad que utiliza solo valores de luma (por ejemplo, valores de intensidad) para identificar la profundidad se proporciona con fines ilustrativos y no debe considerarse limitante. En otros ejemplos, puede utilizarse cualquier técnica para indicar las profundidades relativas de los píxeles en el componente de vista de textura.

De acuerdo con la codificación multivista, los componentes de vista de textura se inter-predicen a partir de componentes de vista de textura en la misma vista o de componentes de vista de textura en una o más vistas diferentes. Los componentes de vista de textura pueden codificarse en bloques de datos de vídeo, que se denominan "bloques de vídeo" y comúnmente llamados macrobloques en el contexto H.264, o bloques arbolados o unidades de codificación (CU) en el contexto HEVC.

Las imágenes de cualquier instancia de tiempo similar pueden incluir contenido similar. Sin embargo, el contenido de vídeo de diferentes imágenes en una instancia de tiempo similar puede estar ligeramente desplazada en la dirección horizontal uno con relación al otro. Por ejemplo, si un bloque se encuentra en (x, y) en la imagen 0 de la vista S0, un bloque situado en $(x+x', y)$ en la imagen 1 de la vista S1 incluye contenido de vídeo similar al bloque situado en (x, y) en la imagen 0 de la vista S0. En este ejemplo, el bloque situado en (x, y) en la imagen 0 de la vista S0 y el bloque situado en $(x + x', y)$ en la imagen 1 de la vista S1 se consideran bloques correspondientes. En algunos ejemplos, un DV para el bloque situado en $(x+x', y)$ en la imagen 1 de la vista S1 se refiere a la ubicación de su bloque correspondiente. Por ejemplo, el DV para el bloque situado en $(x+x', y)$ es $(-x', 0)$.

En algunos ejemplos, el codificador de vídeo 20 o el descodificador de vídeo 30 pueden utilizar el DV de un bloque en una imagen de una primera vista para identificar el bloque correspondiente en una imagen de una segunda vista. El codificador de vídeo 20 y el descodificador de vídeo 20 pueden, por ejemplo, utilizar el DV cuando se realiza una predicción de inter-vista. El codificador de vídeo 20 y el descodificador de vídeo 30 pueden, por ejemplo, realizar una predicción de inter-vista utilizando información de un bloque de referencia de una imagen de referencia en una vista de referencia determinada por el DV del bloque actual.

La FIG. 3 es un diagrama conceptual que ilustra un ejemplo de patrón de predicción temporal y de inter-vista para la codificación de vídeo multivista. De forma similar al ejemplo de la FIG. 2, en el ejemplo de la FIG. 3, se ilustran ocho vistas (con ID de vista "S0" a "S7") y se ilustran doce unidades de acceso o posiciones temporales ("T0" a "T11") para cada vista. Es decir, cada fila de la FIG. 3 corresponde a una vista, mientras que cada columna indica una unidad de acceso o ubicación temporal. Los objetos (que pueden ser imágenes, o bloques de vídeo de ejemplo en imágenes diferentes) se indican en la intersección de cada fila y cada columna de la FIG. 3. La norma H.264/AVC con extensiones MVC puede usar el término trama para representar una porción del vídeo, mientras que la norma HEVC puede usar el término imagen para representar una porción del vídeo. Esta divulgación utiliza el término imagen y trama de forma intercambiable.

En la FIG. 3, la vista S0 puede considerarse como una vista de base, y las vistas S1-S7 pueden considerarse como vistas dependientes. Una vista de base incluye imágenes que no están predichas con inter-vista. Las imágenes en una vista de base pueden inter-predicirse con respecto a otras imágenes en la misma vista. Por ejemplo, ninguna de las imágenes en vista S0 puede inter-predicirse con respecto a una imagen en cualquiera de las vistas S1-S7, pero algunas de las imágenes en la vista S0 pueden inter-predicirse con respecto a otras imágenes en la vista S0.

Además, las unidades de acceso T0 y T8 son unidades de acceso aleatorio o puntos de acceso aleatorio para la secuencia de vídeo de la estructura de predicción de ejemplo de la FIG. 3. Como se ilustra mediante los bloques etiquetados como "I" en la estructura de predicción de ejemplo de la FIG. 3, en cada punto de acceso aleatorio (T0 y T8), los bloques de vídeo de la imagen de vista de base (S0) son predichos intra-imagen. Los bloques de vídeo de otras imágenes de vistas no de base en los puntos de acceso aleatorio, o de imágenes de vista de base y no de base en puntos de acceso no aleatorios, pueden predicirse entre imágenes, ya sea mediante inter-predicción temporal o predicción de inter-vista, como se ilustra mediante los diversos bloques etiquetados como "I", "B", "P" o "b" en la estructura de predicción de ejemplo de la FIG. 3. Las predicciones en la estructura de predicción de ejemplo de la FIG. 3 se indican mediante flechas, donde el objeto al que se apunta utiliza el objeto desde el que se apunta como referencia de predicción.

Una vista dependiente incluye imágenes con predicción de inter-vista. Por ejemplo, cada una de las vistas S1-S7

incluye al menos una imagen que está inter-predicha con respecto a una imagen en otra vista. Las imágenes en una vista dependiente pueden inter-predecirse con respecto a las imágenes en la vista de base, o pueden inter-predecirse con respecto a las imágenes en otras vistas dependientes. En el ejemplo de la FIG. 3, la letra "B" mayúscula y la "b" minúscula se utilizan para indicar diferentes relaciones jerárquicas entre las imágenes, en lugar de diferentes metodologías de codificación. En general, las imágenes con "B" mayúscula son relativamente más altas en la jerarquía de predicción que las tramas con "b" minúscula.

Una secuencia de vídeo que incluye tanto una vista de base como una o más vistas dependientes pueden ser decodificables mediante diferentes tipos de descodificadores de vídeo. Por ejemplo, un tipo básico de descodificador de vídeo puede estar configurado para descodificar solamente la vista de base. Además, puede configurarse otro tipo de descodificador de vídeo para descodificar cada una de las vistas S0 a S7. Un descodificador que está configurado para descodificar tanto la vista de base como las vistas dependientes puede denominarse un descodificador que soporta codificación multivista.

Las imágenes (u otros objetos) de la FIG. 3 se ilustran usando un bloque sombreado que incluye una letra, que designa si la imagen correspondiente está intra-codificada (es decir, una imagen I), inter-codificada en una dirección (es decir, como una imagen P) o inter-codificada 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 utilizan la imagen desde la que se apunta como referencia de predicción. Por ejemplo, la imagen P de la vista S2 en la ubicación temporal T0 se predice a partir de la imagen I de la vista S0 en la ubicación temporal T0.

Al igual que con la codificación de vídeo de vista única, las imágenes de una secuencia de vídeo de codificación de vídeo multivista pueden codificarse predictivamente con respecto a las imágenes en diferentes ubicaciones temporales. Por ejemplo, la imagen B de la vista S0 en la ubicación temporal T1 tiene una flecha apuntando a la misma desde la imagen I de la vista S0 en la ubicación temporal T0, lo cual indica que la imagen b se predice a partir de la imagen I. Adicionalmente, sin embargo, en el contexto de la codificación de vídeo multivista, las imágenes pueden tener predicción de inter-vista. Es decir, un componente de vista (por ejemplo, un componente de vista de textura) puede utilizar los componentes de vista en otras vistas como referencia. En codificación multivista, por ejemplo, la predicción de inter-vista se realiza como si el componente de vista en otra vista fuera una referencia de inter-predicción. Las posibles referencias de inter-vista pueden señalarse, y pueden modificarse mediante el proceso de construcción de la lista de imágenes de referencia, lo cual permite el ordenamiento flexible de las referencias de inter-predicción o de predicción de inter-vista.

La FIG. 3 proporciona varios ejemplos de predicción de inter-vista. Las imágenes de la vista S1, en el ejemplo de la FIG. 3, se ilustran como predichas a partir de imágenes en diferentes ubicaciones temporales de la vista S1, así como con predicción de inter-vista a partir de imágenes de vistas S0 y S2 en las mismas ubicaciones temporales. Por ejemplo, la imagen B de la vista S1 en la ubicación temporal T1 se predice a partir de cada una de las imágenes B de la vista S1 en las ubicaciones temporales T0 y T2, así como las imágenes B de las vistas S0 y S2 en la ubicación temporal T1.

La FIG. 3 también ilustra las variaciones en la jerarquía de predicción utilizando diferentes niveles de sombreado, donde las tramas con una mayor magnitud de sombreado (es decir, relativamente más oscuras) están más altas en la jerarquía de predicción que aquellas tramas que tienen menos sombreado (es decir, relativamente más claras). Por ejemplo, todas las imágenes I en la FIG. 3 se ilustran con sombreado completo, mientras que las imágenes P tienen un sombreado algo más claro, y las imágenes B (y las imágenes con b minúscula) tienen diversos niveles de sombreado, pero siempre más claros que el sombreado de las imágenes P y las imágenes I.

En general, la jerarquía de predicción puede estar relacionada con índices de orden de vista, en que las imágenes relativamente más altas en la jerarquía de predicción deberían descodificarse antes de descodificar imágenes que son relativamente más bajas en la jerarquía. Esas imágenes relativamente más altas en la jerarquía pueden utilizarse como imágenes de referencia durante la descodificación de las imágenes relativamente más bajas en la jerarquía. Un índice de orden de vista es un índice que indica el orden de descodificación de componentes de vista en una unidad de acceso. La descodificación de los componentes de vista puede seguir el orden ascendente del índice de orden de vista. Si se presentan todas las vistas, entonces el conjunto de índices de orden de vista puede comprender un conjunto consecutivamente ordenado de cero a uno menos que el número total de vistas.

Para ciertas imágenes a niveles iguales de la jerarquía, el orden de descodificación entre ellas puede no importar. Por ejemplo, la imagen I de la vista S0 en la posición temporal T0 puede utilizarse como imagen de referencia para la imagen P de la vista S2 en la posición temporal T0, que a su vez puede utilizarse como imagen de referencia para la imagen P de la vista S4 en la posición temporal T0. En consecuencia, la imagen I de la vista S0 en la posición temporal T0 debería descodificarse antes que la imagen P de la vista S2 en la posición temporal T0, que a su vez debería descodificarse antes de la imagen P de la vista S4 en la posición temporal T0. Sin embargo, entre las vistas S1 y S3, no importa un orden de descodificación, porque las vistas S1 y S3 no dependen la una de la otra para la predicción. En su lugar, las vistas S1 y S3 se predicen solo a partir de otras vistas que están más altas en la jerarquía de predicción. Además, la vista S1 puede descodificarse antes que la vista S4, siempre que la vista S1 se descodifique después de las vistas S0 y S2.

Como se ha descrito anteriormente, en 3D-HEVC, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden inter-predecir un bloque actual dentro de una imagen actual de una primera vista con referencia a un bloque de referencia dentro de una imagen de referencia de una segunda vista. Dicha inter-predicción se denomina predicción de inter-vista. La instancia de tiempo de la imagen actual y la imagen de referencia pueden ser las mismas en vistas respectivas. En tales ejemplos, el codificador de vídeo 20 o el decodificador de vídeo 30 realizan predicción de inter-vista a través de imágenes en una misma unidad de acceso, donde las imágenes en la misma unidad de acceso están en la misma instancia de tiempo.

Para realizar una predicción de inter-vista sobre el bloque actual, el codificador de vídeo 20 o el decodificador de vídeo 30 construyen listas de imágenes de referencia que identifican imágenes de referencia que pueden usarse para inter-predicción, incluyendo imágenes que pueden usarse para la predicción de inter-vista. La inter-predicción se refiere a la predicción de un bloque actual en una imagen actual con respecto a un bloque de referencia en una imagen de referencia. La predicción de inter-vista es un subconjunto de inter-predicción en que en la predicción de inter-vista, la imagen de referencia está en una vista diferente a la vista de la imagen actual. Por consiguiente, para la predicción de inter-vista, el codificador de vídeo 20 y el decodificador de vídeo 30 añaden la imagen de referencia en la otra vista en una o ambas listas de imágenes de referencia construidas. La imagen de referencia en la otra vista puede identificarse en cualquier posición dentro de las listas de imágenes de referencia construidas. Tal como se utiliza en esta divulgación, cuando el codificador de vídeo 20 realiza inter-predicción (por ejemplo, inter-prediciendo) en un bloque, el codificador de vídeo 20 puede considerarse como una codificación de inter-predicción de un bloque. Cuando el decodificador de vídeo 30 está realizando una inter predicción (por ejemplo, inter-prediciendo) en un bloque, el decodificador de vídeo 30 puede considerarse como inter-predicción descodificando un bloque. En la predicción de inter-vista, un DMV para el bloque de vídeo actual identifica una ubicación del bloque en una imagen de referencia en una vista distinta a la vista en la que la imagen incluye el bloque de vídeo a predecir que se va a utilizar como bloque de referencia para inter-predecir el bloque actual, y un índice de referencia en una o ambas listas de imágenes de referencia construidas identifica la imagen de referencia en la otra vista.

Esta divulgación describe técnicas para realizar ARP que incluyen identificar un DMV del bloque de vídeo actual o de un bloque de vídeo de referencia y determinar un bloque de predicción residual para el bloque de vídeo actual basándose en el DMV identificado. Un DMV del bloque de vídeo actual, o de un bloque de vídeo de referencia en la misma vista que el bloque de vídeo actual, puede considerarse un DMV desde la vista actual de un bloque de vídeo actual a una vista de referencia que se usa para la predicción de inter-vista de datos de vídeo en la vista actual basándose en datos de vídeo en la vista de referencia. Las técnicas de esta divulgación pueden ser implementadas por uno o ambos del codificador de vídeo 20 y el decodificador de vídeo 30. Estas técnicas pueden ser utilizadas, por ejemplo, en combinación con codificación de vídeo multivista basándose en HEVC y/o codificación de vídeo en 3D basándose en HEVC.

Como se ha analizado anteriormente, los datos que definen un TMV o DMV para un bloque de datos de vídeo pueden incluir componentes horizontales y verticales del vector, así como una resolución para el vector. La información de movimiento para un bloque de vídeo puede incluir un vector de movimiento, así como una dirección de predicción y un valor de índice de imagen de referencia. Además, la información de movimiento para un bloque de vídeo actual puede predecirse a partir de la información de movimiento de un bloque de vídeo contiguo que también se puede denominar un bloque de vídeo de referencia. El bloque de vídeo de referencia puede ser un bloque contiguo espacial dentro de la misma imagen, un bloque contiguo temporal dentro de una imagen diferente de la misma vista, pero dentro de una unidad de acceso diferente, o un bloque de vídeo dentro de una imagen diferente de una vista diferente, pero dentro de la misma unidad de acceso. En el caso de la información de movimiento de un bloque de referencia en una vista diferente, el vector de movimiento puede ser un TMV derivado de un bloque de referencia en una imagen de referencia de inter-vista (es decir, una imagen de referencia en la misma unidad de acceso que la imagen actual, pero desde una perspectiva diferente), o un DMV derivado de un DV.

Típicamente, para la predicción de información de movimiento, se forma una lista de información de movimiento candidatos a partir de varios bloques de referencia de una manera definida, por ejemplo, de tal manera que la información de movimiento de varios bloques de referencia se considera para su inclusión en la lista en un orden definido. Después de formar la lista de candidatos, el codificador de vídeo 20 puede evaluar cada candidato para determinar cuál proporciona las mejores características de velocidad y distorsión que mejor coincidan con un perfil de velocidad y distorsión dado seleccionados para codificar el vídeo. El codificador de vídeo 20 puede realizar un procedimiento de optimización de distorsión de velocidad (RDO) con respecto a cada uno de los candidatos, seleccionando uno de los candidatos de información de movimiento que tienen los mejores resultados de RDO. De forma alternativa, el codificador de vídeo 20 puede seleccionar uno de los candidatos almacenados en la lista que mejor se aproxime a la información de movimiento determinada para el bloque de vídeo actual.

En cualquier caso, el codificador de vídeo 20 puede especificar el candidato seleccionado utilizando un índice que identifica al candidato seleccionado de la lista de candidatos de información de movimiento. El codificador de vídeo 20 puede señalar este índice en el flujo de bits codificado para su uso por parte del decodificador de vídeo 30. Para la eficiencia de codificación, los candidatos pueden ordenarse en la lista de tal manera que la información de movimiento de candidatos que más probablemente sea seleccionada para codificar el bloque de vídeo actual es la

primera, o de lo contrario está asociada con el valor de índice de magnitud más bajo.

Las técnicas para la predicción de información de movimiento pueden incluir un modo de fusión, un modo de omisión y un modo de predicción de vector de movimiento anticipado (AMVP). En general, de acuerdo con el modo de fusión y/o modo de omisión, un bloque de vídeo actual hereda la información de movimiento, por ejemplo, vector de movimiento, dirección de predicción e índice de imagen de referencia, de otro bloque contiguo previamente codificado, por ejemplo, un bloque espacialmente contiguo en la misma imagen, o un bloque en una imagen de referencia temporal o inter-vista. Cuando se implementa el modo de fusión / omisión, el codificador de vídeo 20 construye una lista de candidatos de fusión que son la información de movimiento de los bloques de referencia en una materia definida, selecciona uno de los candidatos de fusión y señala un índice de lista de candidatos que identifica al candidato de fusión seleccionado para descodificador de vídeo 30 en el flujo de bits.

El descodificador de vídeo 30, al implementar el modo de fusión / omisión, recibe este índice de lista de candidatos, reconstruye la lista de candidatos de fusión de acuerdo con la manera definida y selecciona uno de los candidatos de fusión en la lista de candidatos indicada por el índice. El descodificador de vídeo 30 puede entonces instanciar el candidato seleccionado de los candidatos de fusión como un vector de movimiento para la PU actual con la misma resolución que el vector de movimiento del candidato seleccionado de los candidatos de fusión y apuntando a la misma imagen de referencia que el vector de movimiento para el seleccionado de los candidatos de fusión. Por consiguiente, en el lado del descodificador, una vez descodificado el índice de lista de candidatos, se puede heredar toda la información de movimiento del bloque correspondiente del candidato seleccionado tal como, por ejemplo, vector de movimiento, dirección de predicción e índice de imagen de referencia. El modo de fusión y el modo de omisión promueven la eficiencia del flujo de bits permitiendo que el codificador de vídeo 20 señale un índice en la lista de candidatos de fusión, en lugar de toda la información de movimiento para la inter-predicción del bloque de vídeo actual.

Cuando se implementa AMVP, el codificador de vídeo 20 construye una lista de predictores de vector de movimiento candidatos (MVP) en una materia definida, selecciona uno de los MVP candidatos y señala un índice de lista de candidatos que identifica el MVP seleccionado con el descodificador de vídeo 30 en el flujo de bits. Similar al modo de fusión, al implementar AMVP, el descodificador de vídeo 30 reconstruye la lista de MVP candidatos en el asunto definido, descodifica el índice de lista de candidatos del codificador y selecciona e instancia uno de los MVP basándose en el índice de lista de candidatos.

Sin embargo, contrariamente al modo de fusión / omisión, al implementar AMVP, el codificador de vídeo 20 también señala un índice de imagen de referencia y una dirección de predicción, especificando así la imagen de referencia a la que apunta el MVP especificado por los puntos del índice de lista de candidatos. Además, el codificador de vídeo 20 determina una diferencia de vector de movimiento (MVD) para el bloque actual, donde el MVD es una diferencia entre el MVP y el vector de movimiento real que de otro modo se utilizaría para el bloque actual. Para AMVP, además del índice de imagen de referencia, la dirección de la imagen de referencia y el índice de lista de candidatos, el codificador de vídeo 20 señala el MVD para el bloque actual en el flujo de bits. Debido a la señalización del índice de imagen de referencia y la diferencia de vector de predicción para un bloque dado, el AMVP puede no ser tan eficiente como el modo de fusión / omisión, pero puede proporcionar una fidelidad mejorada de los datos de vídeo codificados.

La FIG. 4 muestra un ejemplo de un bloque de vídeo actual 47, cinco bloques contiguos espaciales (41, 42, 43, 44 y 45) y un bloque de referencia temporal 46 de otra imagen pero en la misma vista que la imagen actual. El bloque de referencia temporal 46 puede ser, por ejemplo, un bloque co-ubicado en una imagen de una instancia temporal diferente, pero la misma vista que el bloque de vídeo actual 47. En algunos ejemplos, el bloque de vídeo actual 47 y los bloques de vídeo de referencia 41-46 pueden definirse en general en la norma HEVC actualmente en desarrollo. Los bloques de vídeo de referencia 41-46 se denominan A0, A1, B0, B1, B2 y T de acuerdo con la norma HEVC actualmente en desarrollo. El codificador de vídeo 20 y el descodificador de vídeo 30 pueden predecir la información de movimiento, incluyendo un TMV, del bloque de vídeo actual 47 basándose en la información de movimiento de los bloques de vídeo de referencia 41-46 de acuerdo con un modo de predicción de información de movimiento, por ejemplo, modo de fusión / omisión o modo AMVP. Como se describe con más detalle a continuación, los TMV de bloques de vídeo se pueden usar con DMV para predicción residual avanzada de acuerdo con las técnicas de esta divulgación.

Como se ilustra en la FIG. 4, los bloques de vídeo 42, 44, 43, 41 y 45 pueden quedar a la izquierda, por encima, por encima a la derecha, por debajo a la izquierda y por encima a la izquierda, respectivamente, con relación al bloque de vídeo actual 47. Sin embargo, el número y las ubicaciones de los bloques contiguos 41-45 con respecto al bloque de vídeo actual 47 ilustrado en la FIG. 4, son simplemente un ejemplos. En otras ubicaciones, la información de movimiento de un número diferente de bloques contiguos y/o de bloques en ubicaciones diferentes, puede considerarse para su inclusión en una lista de candidatos de predicción de información de movimiento para el bloque de vídeo actual 47.

La relación espacial de cada uno de los bloques espacialmente contiguos 42, 44, 43, 41 y 45 con el bloque de vídeo actual 47 se puede describir de la forma siguiente. Se utiliza una ubicación de luma (xP, yP) para especificar la

muestra de luma superior izquierda del bloque actual en relación con la muestra superior izquierda de la imagen actual. Las variables nPSW y nPSH indican la anchura y la altura del bloque actual para luma. La muestra de luma superior izquierda del bloque 42 espacialmente contiguo es $xP - 1, yP + nPSH - 1$. La muestra de luma superior izquierda del bloque 44 espacialmente contiguo es $xP + nPSW - 1, yP - 1$. La muestra de luma superior izquierda del bloque espacialmente contiguo 43 es $xP + nPSW, yP - 1$. La muestra de luma superior izquierda del bloque espacialmente contiguo 41 es $xP - 1, yP + nPSH$. La muestra de luma superior izquierda del bloque espacialmente contiguo 45 es $xP - 1, yP - 1$. Aunque se describen con respecto a las ubicaciones de luma, los bloques actuales y de referencia pueden incluir componentes de croma.

- 10 Cada uno de los bloques 41-45 espacialmente contiguos puede proporcionar un candidato de información de movimiento espacial para predecir la información de movimiento, por ejemplo, TMV, del bloque de vídeo actual 47. Un codificador de vídeo, por ejemplo, el codificador de vídeo 20 (FIG. 1) o el decodificador de vídeo 30 (FIG. 1), puede considerar la información de movimiento de los bloques de referencia espacialmente contiguos en un orden predeterminado, por ejemplo, un orden de escaneado. En el caso de 3D-HEVC, por ejemplo, el decodificador de vídeo puede considerar la información de movimiento de los bloques de referencia para su inclusión en una lista de candidatos de fusión para el modo de fusión en el siguiente orden: 42, 44, 43, 41 y 45. En el ejemplo ilustrado, los bloques espacialmente contiguos 41-45 están a la izquierda de y/o encima del bloque de vídeo actual 47. Esta disposición es típica, ya que la mayoría de los codificadores de vídeo codifican bloques de vídeo en orden de escaneado de trama desde la parte superior izquierda de una imagen. Por consiguiente, en dichos ejemplos, los bloques espacialmente contiguos 41 - 45 se codificarán típicamente antes del bloque de vídeo actual 47. Sin embargo, en otros ejemplos, por ejemplo, cuando un codificador de vídeo codifica bloques de vídeo en un orden diferente, los bloques 41-45 espacialmente contiguos pueden estar situados a la derecha de y/o debajo del bloque de vídeo actual 47.
- 25 El bloque de referencia temporal 46 está situado dentro de una imagen de referencia temporal codificada previamente, aunque no necesariamente inmediatamente antes en orden de codificación, a la imagen actual del bloque de vídeo actual 47. Además, la imagen de referencia del bloque 46 no es necesariamente anterior a la imagen del bloque de vídeo actual 47 en orden de vista. El bloque de vídeo de referencia 46 puede estar co-ubicado, en general, en la imagen de referencia con respecto a la ubicación del bloque de vídeo actual 47 en la imagen actual. En algunos ejemplos, el bloque de vídeo de referencia 46 está situado a la derecha de y por debajo de la posición del bloque de vídeo actual 47 en la imagen actual o cubre la posición central del bloque de vídeo actual 47 en la imagen actual.

- 35 La FIG. 5 es un diagrama conceptual que ilustra un ejemplo de obtención de un candidato de vector de movimiento predicho de inter-vista (IPMVC) y un candidato de vector de movimiento de disparidad de inter-vista (IDMVC) para la predicción de la información de movimiento de un bloque de vídeo actual 50, por ejemplo, de acuerdo con un modo de fusión / omisión o un modo AMVP. Cuando la predicción de inter-vista está habilitada, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden añadir un nuevo candidato de vector de movimiento, IPMVC o IDMVC a la lista de candidatos de información de movimiento para el bloque de vídeo actual 50. El IPMVC puede predecir un TMV para el bloque de vídeo actual 50, que el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden usar para ARP del bloque de vídeo actual 50 u otro bloque de vídeo de acuerdo con las técnicas de esta divulgación, como se describe con mayor detalle a continuación. El IDMVC puede predecir un DMV para el bloque de vídeo actual 50, que el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden usar para ARP del bloque de vídeo actual 50 u otro bloque de vídeo de acuerdo con las técnicas de esta divulgación, como se describe con mayor detalle a continuación.

- 50 En el ejemplo de la FIG. 5, el bloque actual 50 está en la vista actual V_m . El codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden localizar un bloque 52 correspondiente o de referencia en la vista de referencia V_0 usando un vector de disparidad (DV) 51. El codificador de vídeo puede determinar DV 51 basándose en parámetros de cámara, o de acuerdo con cualquiera de las técnicas descritas en el presente documento. Por ejemplo, el codificador de vídeo puede determinar DV 51 para el bloque de vídeo actual 50 incluyendo basándose en un DV de un bloque contiguo, por ejemplo, usando Obtención de Vector de Disparidad Basándose en Bloques Contiguos (NBDV).

- 55 Si el bloque de referencia 52 no está intracodificado y no tiene predicción de inter-vista, y su imagen de referencia, por ejemplo, la imagen de referencia 58 o la imagen de referencia 60, tiene un valor de recuento de orden de imagen (POC) igual al de una entrada en la misma lista de imágenes de referencia del bloque de vídeo actual 50, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden obtener su información de movimiento (dirección de predicción, imágenes de referencia y vectores de movimiento), después de convertir el índice de referencia basándose en POC, para ser el IPMVC para el bloque de vídeo actual 50. En el ejemplo de la FIG. 5, el bloque de vídeo de referencia 52 está asociado con TMV 54 que apunta a una primera imagen de referencia 58 en la vista de referencia V_0 que se especifica en una primera lista de imágenes de referencia (RefPicList0) y TMV 56 que apunta a una segunda imagen 60 en la vista de referencia V_0 que se especifica en una segunda lista de imágenes de referencia (RefPicList1). La herencia de los TMV 54 y 56 por el bloque de vídeo actual 50 se ilustra mediante las flechas discontinuas en la FIG.5. Basándose en la información de movimiento del bloque de vídeo de referencia 52, el codificador de vídeo obtiene el IPMVC para que el bloque de vídeo actual 50 sea al menos uno de un TMV 62 que

apunte a una primera imagen de referencia 66 en la vista actual Vm que se especifica en una primera lista de imágenes de referencia (RefPicList0), por ejemplo, con el mismo POC en la primera lista de imágenes de referencia como imagen de referencia 58 y TMV 64 apuntando a una segunda imagen 68 en la vista actual Vm que se especifica en una segunda lista de imágenes de referencia (RefPicList1) Por ejemplo, con el mismo POC que la imagen de referencia 60. El codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden usar TMV 62 y/o TMV 64 para ARP del bloque de vídeo actual 50 u otro bloque de vídeo de acuerdo con las técnicas de esta divulgación, como se describe con mayor detalle a continuación.

El codificador de vídeo 20 y/o decodificador de vídeo 30 también pueden convertir DV 51 a un IDMVC para el bloque de vídeo actual 50 y añadir el IDMVC a la lista de candidatos de información de movimiento para el bloque de vídeo actual 50 en una posición diferente de la IPMVC. Cada uno de los IPMVC o IDMVC puede denominarse un "candidato de inter-vista" en este contexto. En el modo de fusión / omisión, el codificador de vídeo inserta el IPMVC, si está disponible, antes de todos los candidatos de fusión espacial y temporal en la lista de candidatos de fusión. En el modo de fusión / omisión, el codificador de vídeo inserta el IDMVC antes del candidato de fusión espacial derivado de A0 (bloque 41 de la FIG. 4). La conversión de DV 51 a un IDMVC puede considerarse conversión de DV 51 a un DMV para el bloque de vídeo actual 50. El codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden usar el DMV para ARP del bloque de vídeo actual 50 u otro bloque de vídeo de acuerdo con las técnicas de esta divulgación, como se describe con mayor detalle a continuación.

En algunas situaciones, un codificador de vídeo puede obtener un DV para un bloque de vídeo actual. Por ejemplo, como se ha descrito anteriormente, con referencia a la FIG. 5, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden obtener DV 51 para el bloque de vídeo actual 50. En algunos ejemplos, un codificador de vídeo puede utilizar la obtención de Vector de Disparidad Basándose en Bloques Contiguos (NBDV) para obtener el DV para el bloque de vídeo actual.

Las propuestas para 3D-HEVC usan un orden de codificación de textura primero para todas las vistas. En otras palabras, para cada una de la pluralidad de vistas en el flujo de bits, el componente de textura de está codificado, por ejemplo, codificado o decodificado, antes de cualquier componente de profundidad para la vista. En algunos casos, por ejemplo, para la predicción de inter-vista, se necesita un DV para codificar un bloque de vídeo en un componente de textura de una vista en una unidad de acceso particular. Sin embargo, en la codificación de texto primero, el componente de profundidad correspondiente del bloque de vídeo actual no está disponible para determinar el DV para el bloque de vídeo actual. El NBDV puede ser empleado por un codificador de vídeo, y se propone para 3D-HEVC, para obtener un DV para un bloque de vídeo actual en tales situaciones. En el diseño actual de 3D-HEVC, el DV derivado de NBDV podría refinarse adicionalmente recuperando los datos de profundidad del mapa de profundidad de la vista de referencia apuntados por el DV desde el proceso NBDV.

Un DV se utiliza para un estimador del desplazamiento entre dos vistas. Debido a que los bloques contiguos comparten casi la misma información de movimiento / disparidad en la codificación de vídeo, el bloque de vídeo actual puede usar la información de vector de movimiento en bloques contiguos como un buen predictor de su información de movimiento / disparidad. Siguiendo esta idea, NBDV utiliza la información de disparidad contigua para estimar los DV en diferentes vistas.

De acuerdo con NBDV, un codificador de vídeo identifica varios bloques contiguos espaciales y temporales. Se utilizan dos conjuntos de bloques contiguos. Un conjunto es de bloques contiguos espaciales y el otro conjunto es de bloques contiguos temporales. El codificador de vídeo comprueba cada uno de los bloques contiguos espaciales y temporales en un orden predefinido determinado por la prioridad de la correlación entre el bloque actual y el bloque candidato (contiguo). Cuando el codificador de vídeo identifica un DMV, es decir, un vector de movimiento que apunta desde el bloque candidato contiguo a una imagen de referencia de inter-vista (en la misma unidad de acceso, pero en una vista diferente), en la información de movimiento de los candidatos, el codificador de vídeo convierte el DMV a un DV, y devuelve el índice de orden de vista asociado. Por ejemplo, el codificador de vídeo puede establecer un componente horizontal del DV para el bloque actual igual a un componente horizontal del DMV y puede establecer el componente vertical del DV en 0.

3D-HEVC adoptó en primer lugar el procedimiento NBDV propuesto en Zhang et al. "3D-CE5.h: Resultados de la generación de vectores de disparidad", Equipo de Colaboración Conjunta en el Desarrollo de la Extensión de Codificación de Vídeo de la ITU-T SG 16 WP 3 e ISO/IEC JTC1/SC 29/WG 11, 1.ª Reunión: Estocolmo, Suecia, 16-20 de julio de 2012, Doc. JCT3V-A0097 (número MPEG m26052), en lo sucesivo "JCT3V-A0097". JCT3V-A0097 se puede descargar desde el siguiente enlace:

http://phenix.int-evry.fr/jct2/doc_end_user/current_document.php?id=89.

En algunos diseños de 3D-HEVC, cuando el codificador de vídeo realiza el proceso NBDV, el codificador de vídeo comprueba los vectores de movimiento de disparidad en los bloques contiguos temporales, los vectores de movimiento de disparidad en los bloques contiguos espaciales y luego los vectores de disparidad implícitos (IDVs) en orden. Un IDV puede ser un vector de disparidad de una PU espacialmente o temporalmente contigua que se codifica utilizando la predicción de inter-vista. Los IDV también pueden denominarse vectores de disparidad

derivados. Un IDV puede generarse cuando una PU emplea la predicción de inter-vista, es decir, el candidato para AMVP o modos de fusión se obtiene a partir de un bloque de referencia en la otra vista con la ayuda de un vector de disparidad. Este vector de disparidad se llama IDV. Un IDV puede almacenarse en la PU con el fin de obtención DV. Por ejemplo, aunque el bloque se codifica con predicción de movimiento, un DV derivado para el bloque no se descarta con el propósito de codificar un bloque de vídeo siguiente. Por lo tanto, cuando el codificador de vídeo identifica un DMV o un IDV, el codificador de vídeo puede devolver el DMV o IDV identificado.

Se incluyeron vectores de disparidad implícitos (IDVs) con un NBDV simplificado en Sung et al., "3D-CE5.h: Simplificación de la obtención del vector de disparidad para la codificación de vídeo en 3D basándose en HEVC", Equipo de Colaboración Conjunta en el Desarrollo de la Extensión de Codificación de Vídeo de la ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 1.^a Reunión: Estocolmo, Suecia, 16-20 de julio de 2012, Doc. JCT3V-A0126 (número MPEG m26079), en lo sucesivo "JCT3V-A0126". JCT3V-A0126 se puede descargar desde el siguiente enlace:

http://phenix.int-evry.fr/jct2/doc_end_user/current_document.php?id=142.

El desarrollo adicional del NBDV para 3D-HEVC se produjo en Kang et al., "3D-CE5.h: Mejora de la obtención de los vectores de disparidad", Equipo de Colaboración Conjunta en el Desarrollo de la Extensión de Codificación de Vídeo de la ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 2.^a Reunión: Shanghai, China, 13-19 de octubre de 2012, Doc. JCT3V-B0047 (número MPEG m26736), en lo sucesivo "JCT3V-B0047". JCT3V-B0047 se puede descargar desde el siguiente enlace:

http://phenix.it-sudparis.eu/jct2/doc_end_user/current_document.php?id=236. En JCT3V-B0047, el NBDV for3D-HEVC se simplificó adicionalmente eliminando los IDV almacenados en la memoria intermedia de imágenes descodificadas, pero la ganancia de codificación también se mejora con la selección de imagen de punto de acceso aleatorio (RAP). El codificador de vídeo puede convertir el IDV o vector de movimiento de disparidad devuelto en un vector de disparidad y puede utilizar el vector de disparidad para la predicción de inter-vista y la predicción residual de inter-vista. El acceso aleatorio se refiere a la descodificación de un flujo de bits a partir de una imagen codificada que no es la primera imagen codificada en el flujo de bits. La inserción de imágenes de acceso aleatorio o puntos de acceso aleatorio en un flujo de bits a intervalos regulares puede permitir el acceso aleatorio. Entre los tipos de ejemplo de imágenes de acceso aleatorio se incluyen imágenes de actualización instantánea de descodificador instantánea (IDR), imágenes de acceso aleatorio limpio (CRA) e imágenes de acceso de enlace roto (BLA). Por lo tanto, las imágenes IDR, CRA y BLA se denominan colectivamente imágenes RAP. En algunos ejemplos, las imágenes RAP pueden tener tipos de unidades NAL iguales a BLA_W_LP, BLA_W_RADL, BLA_N_LP, IDR_W_RADL, IDR_N_LP, RSV_IRAP_VCL22, RSV_IRAP_VCL23, o CRA_NUT.

Las técnicas para la obtención DV basándose en CU para 3D-HEVC se propusieron en Kang et al., "CE2.h: Obtención de Vector de Disparidad basándose en CU en 3D-HEVC", Equipo de Colaboración Conjunta en el Desarrollo de la Extensión de Codificación de Vídeo de la ITU-T SG 16 WP3 e ISO/IEC JTC 1/SC 29/WG 11, 4.^a Reunión: Incheon, Corea, 20-26 de abril de 2013, Doc. JCT3V-D0181 (número MPEG m29012), en lo sucesivo "JCT3V-D0181". JCT3V-D0181 se puede descargar desde el siguiente enlace:

http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=866.

Cuando el codificador de vídeo identifica un DMV o un IDV, el codificador de vídeo puede terminar el proceso de comprobación. Así, una vez que el codificador de vídeo encuentra un DV para el bloque actual, el codificador de vídeo puede terminar el proceso NBDV. Cuando el codificador de vídeo no puede determinar un DV para el bloque actual realizando el proceso NBDV (es decir, cuando no se encuentra DMV o IDV durante el proceso NBDV), el NBDV está marcado como no disponible. En otras palabras, puede considerarse que el proceso NBDV devuelve un vector de disparidad no disponible.

Si el codificador de vídeo no puede obtener un DV para el bloque actual (es decir, si no se encuentra ningún vector de disparidad) realizando el proceso NBDV, el codificador de vídeo puede usar un DV cero como DV para la PU actual. El DV cero es un DV que tiene componentes horizontales y verticales iguales a 0. Por lo tanto, incluso cuando el proceso NBDV devuelve un resultado no disponible, otros procesos de codificación del codificador de vídeo que requieren un DV pueden utilizar un vector de disparidad cero para el bloque actual. En algunos ejemplos, si el codificador de vídeo es incapaz de obtener un DV para el bloque actual realizando el proceso NBDV, el codificador de vídeo puede deshabilitar la predicción residual de inter-vista para el bloque actual. Sin embargo, independientemente de si el codificador de vídeo es capaz de obtener un DV para el bloque actual realizando el proceso NBDV, el codificador de vídeo puede usar la predicción de inter-vista para el bloque actual. Es decir, si no se encuentra un DV después de comprobar todos los bloques contiguos predefinidos, se puede utilizar un vector de disparidad cero para la predicción de inter-vista mientras que la predicción residual de inter-vista puede estar deshabilitada para la CU correspondiente.

La FIG. 6 es un diagrama conceptual que ilustra ejemplos de bloques contiguos espaciales, a partir de los cuales se puede obtener un DV para un bloque de vídeo actual utilizando NBDV, con relación al bloque de vídeo actual 90.

Los cinco bloques contiguos espaciales ilustrados en la FIG. 6 son el bloque inferior izquierdo 96, el bloque izquierdo 95, el bloque superior derecho 92, el bloque superior 93 y el bloque superior izquierdo 94 con relación al bloque de vídeo actual. Los bloques contiguos espaciales pueden ser los bloques inferior izquierdo, izquierdo, superior-derecho, superior y superior izquierdo de un bloque de vídeo actual que cubre CU. Debe observarse que estos bloques contiguos espaciales para NBDV pueden ser los mismos que los bloques contiguos espaciales utilizados por el codificador de vídeo para la predicción de información de movimiento para el bloque de vídeo actual, tal como de acuerdo con los modos MERGE / AMVP en HEVC. En tales casos, no se requiere ningún acceso adicional a la memoria por parte del codificador de vídeo para NBDV, ya que la información de movimiento de los bloques contiguos espaciales ya se considera para la predicción de información de movimiento para el bloque de vídeo actual.

Para comprobar bloques contiguos temporales, un codificador de vídeo construye una lista de imágenes candidatas. En algunos ejemplos, el codificador de vídeo puede tratar hasta dos imágenes de referencia desde la vista actual, es decir, la misma vista que el bloque de vídeo actual, como imágenes candidatas. El codificador de vídeo puede insertar primero la imagen de referencia co-ubicada en la lista de imágenes candidatas, seguida por el resto de imágenes candidatas en el orden ascendente del índice de imágenes de referencia. Cuando estén disponibles imágenes de referencia con el mismo índice de referencia en ambas listas de imágenes de referencia, el codificador de vídeo puede insertar la de la misma lista de imágenes de referencia que la imagen co-ubicada antes de la otra imagen de referencia de la otra lista de imágenes de referencia. En algunos ejemplos, el codificador de vídeo puede identificar tres regiones candidatas para obtener los bloques contiguos temporales de cada una de las imágenes candidatas en la lista de imágenes candidatas. Las tres regiones candidatas pueden definirse de la forma siguiente:

- CPU: Una región co-ubicada de la PU actual o la CU actual.
- CLCU: Una unidad de codificación más grande (LCU) que cubre la región co-ubicada del bloque actual.
- BR: Un bloque inferior derecho 4×4 de la CPU.

Si la PU que cubre la región candidata especifica un DMV, el codificador de vídeo puede determinar el DV de la unidad de vídeo actual basándose en el vector de movimiento de disparidad de la PU.

Como se analizó anteriormente, además de los DMV derivados de bloques contiguos espaciales y temporales, un codificador de vídeo puede comprobar IDVs. En el proceso NBDV propuesto para 3D-HTM 7.0, el codificador de vídeo comprueba DMVs en los bloques contiguos temporales, luego los DMVs en los bloques contiguos espaciales, y luego los IDVs, en orden. Una vez que se encuentra el DMV o IDV, se termina el proceso.

Cuando el codificador de vídeo comprueba una PU contigua (es decir, una PU espacialmente o temporalmente contigua), el codificador de vídeo puede comprobar primero si la PU contigua tiene un vector de movimiento de disparidad. Si ninguna de las PU contiguas tiene un vector de movimiento de disparidad, el codificador de vídeo puede determinar si cualquiera de las PU espacialmente contiguas tiene un IDV. Si una de las PU espacialmente contiguas tiene un IDV y el IDV está codificado como modo de fusión / omisión, el codificador de vídeo puede terminar el proceso de comprobación y puede usar el IDV como vector de disparidad final para la PU actual.

Como se indicó anteriormente, un codificador de vídeo puede aplicar un proceso NBDV para obtener un DV para un bloque actual (por ejemplo, una CU, PU, etc.). El vector de disparidad para el bloque actual puede indicar una ubicación en una imagen de referencia (es decir, un componente de referencia) en una vista de referencia. En algunos diseños 3D-HEVC, el codificador de vídeo puede acceder a la información de profundidad para la vista de referencia. En algunos diseños 3D HEVC, cuando el codificador de vídeo utiliza el proceso NBDV para obtener el DV para el bloque actual, el codificador de vídeo puede aplicar un proceso de refinamiento para refinar aún más el vector de disparidad para el bloque actual. El codificador de vídeo puede refinar el DV para el bloque actual basándose en el mapa de profundidad de la imagen de referencia. El codificador de vídeo puede usar un proceso de refinamiento similar para refinar un DMV para la predicción de síntesis de vista hacia atrás. De esta manera, la profundidad se puede usar para refinar el DV o DMV que se utilizará para la predicción de síntesis de vista hacia atrás. Este proceso de refinamiento puede denominarse en este documento refinamiento de NBDV ("NBDV-R"), el proceso de refinamiento de NBDV, o NBDV orientado a la profundidad (Do-NBDV).

Cuando el proceso NBDV devuelve un vector de disparidad disponible (por ejemplo, cuando el proceso NBDV devuelve una variable que indica que el proceso NBDV pudo obtener un vector de disparidad para el bloque actual basándose en un vector de movimiento de disparidad o un IDV de un bloque contiguo), el codificador de vídeo puede refinar aún más el vector de disparidad recuperando los datos de profundidad del mapa de profundidad de la vista de referencia. En algunos ejemplos, el proceso de refinamiento incluye los dos pasos siguientes:

- 1) Localizar un bloque de profundidad correspondiente mediante el DV derivado en la vista de profundidad de referencia previamente codificada, tal como la vista de base; el tamaño del bloque de profundidad correspondiente es el mismo que el de la PU actual.

2) Seleccionar un valor de profundidad de cuatro píxeles de esquina del correspondiente bloque de profundidad y convertirlo en el componente horizontal del DV refinado. El componente vertical del DV no cambia.

El DV refinado puede utilizarse para la predicción de inter-vista para el bloque de vídeo actual, mientras que el DV no refinado puede usarse para la predicción residual de inter-vista para el bloque de vídeo actual. Además, el DV refinado se almacena como el vector de movimiento de una PU si está codificado con el modo de predicción de síntesis de vista atrás (BVSP), que se describe con mayor detalle a continuación. En el proceso de NBDV propuesto para 3D-HTM 7.0, siempre se accederá al componente de vista de profundidad de la vista de base independientemente del valor del índice de orden de vista derivado del proceso NBDV.

Un enfoque de Predicción de Síntesis de Visión hacia Atrás (BVSP) fue propuesto en Tian et al., "CE1.h: Predicción de Síntesis de Vista hacia Atrás Utilizando Bloques Contiguos". Equipo de Colaboración Conjunta en el Desarrollo de la Extensión de Codificación de Vídeo de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 3.ª Reunión: Ginebra, Suiza, 17-23 de enero de 2013, Doc. JCT3VC0152 (número MPEG m27909), en lo sucesivo "JCT3V-C0152". JCT3V-C0152 se puede descargar desde el siguiente enlace: http://phenix.int-evry.fr/jct3v/doc_end_user/current_document.php?id=594.

JCT3V-C0152 fue adoptado en la 3.ª reunión de JCT-3V. La idea básica de este BSVP es la misma que la VSP basándose en bloques en 3D-AVC. Ambas técnicas usan la deformación hacia atrás y la VSP basándose en bloques para evitar transmitir las diferencias de vector de movimiento y utilizar vectores de movimiento más precisos. Los detalles de implementación son diferentes debido a las diferentes plataformas. En los párrafos siguientes, también usamos el término BVSP para indicar uno o ambos del enfoque VSP de deformación hacia atrás en 3D-HEVC o el VSP basándose en bloques en 3D-AVC.

En 3D-HTM, la primera codificación de textura se aplica en condiciones de prueba comunes. Por lo tanto, la vista de profundidad no de base correspondiente no está disponible al descodificar una vista de textura no de base. Por lo tanto, la información de profundidad se estima y se utiliza para realizar BVSP.

En general, cuando un codificador de vídeo realiza BVSP para sintetizar una imagen de textura de referencia, el codificador de vídeo procesa bloques (por ejemplo, unidades de vídeo) en una imagen de textura dependiente. La imagen de textura dependiente y la imagen de textura sintetizada están en la misma unidad de acceso, pero están en vistas diferentes. Cuando el codificador de vídeo procesa un bloque (es decir, un bloque actual) de la imagen de textura dependiente, el codificador de vídeo puede realizar un proceso NBDV para identificar un DV del bloque actual. Es decir, para estimar la información de profundidad para un bloque, un codificador de vídeo puede obtener primero un DV de bloques contiguos.

Además, cuando el codificador de vídeo realiza BVSP para sintetizar la imagen de textura de referencia, el codificador de vídeo puede utilizar el DV del bloque actual para identificar un bloque de referencia en una imagen de profundidad de referencia. En otras palabras, el codificador de vídeo puede utilizar entonces el DV derivado para obtener un bloque de profundidad a partir de una vista de referencia. Por ejemplo, el DV identificado mediante el proceso NBDV se puede denotar como (dvx, dvy) y la posición de bloque actual se puede denotar como (blockx, blocky). Además, en este ejemplo, el codificador de vídeo puede obtener un bloque de profundidad en (blockx+dvx, blocky+dvy) en la imagen en profundidad de la vista de referencia. En este ejemplo, el bloque de profundidad buscado tiene el mismo tamaño de la PU actual. La imagen de textura dependiente y la imagen de profundidad de referencia están en la misma unidad de acceso, pero están en vistas diferentes. El codificador de vídeo puede entonces realizar un proceso de deformación hacia atrás para determinar, basándose en valores de muestra del bloque actual y valores de muestra del bloque de referencia identificado de la imagen de referencia, valores de muestra de la imagen sintetizada. En otras palabras, el codificador de vídeo, en este ejemplo, puede usar el bloque de profundidad buscado para realizar deformación hacia atrás para la PU actual.

Como se ha indicado anteriormente, cuando el codificador de vídeo realiza BVSP, el codificador de vídeo puede realizar un proceso NBDV para identificar un DV para un bloque actual. Además, cuando el codificador de vídeo realiza BVSP, el codificador de vídeo puede usar un proceso de refinamiento similar al descrito en otra parte de esta divulgación para refinar el DMV obtenido usando el proceso NBDV. Cuando el codificador de vídeo realiza el proceso de refinamiento de DV, el codificador de vídeo puede refinar el DV basándose en valores de profundidad en un mapa de profundidad en una vista de referencia. En otras palabras, la profundidad se puede utilizar para refinar el DV o DMV que se utilizará para BVSP. El DV refinado puede almacenarse como el vector de movimiento de una PU si el DV refinado está codificado con el modo BVSP.

En algunas versiones de 3D-HEVC, se aplica la primera codificación de textura. En la primera codificación de textura, un codificador de vídeo codifica (por ejemplo, codifica o descodifica) un componente de vista de textura antes de codificar el componente de vista de profundidad correspondiente (es decir, el componente de vista de profundidad que tiene el mismo valor POC y el identificador de vista como el componente de vista de textura). Por lo tanto, un componente de vista de profundidad de vista no de base no está disponible para su uso en la codificación de un correspondiente componente de vista de textura de vista no de base. En otras palabras, cuando un codificador de vídeo codifica un componente de vista de textura no de base, el componente correspondiente de vista de

profundidad no de base no está disponible. Por lo tanto, la información de profundidad se puede estimar y utilizar para realizar BVSP.

La FIG. 7 es un diagrama conceptual que ilustra la obtención del bloque de profundidad a partir de una vista de referencia para realizar la predicción BVSP. En el ejemplo de la FIG. 7, un codificador de vídeo está codificando una imagen de textura actual 70. La imagen de textura actual 70 está marcada como una "imagen de textura dependiente" porque la imagen de textura actual 70 depende de una imagen de textura de referencia sintetizada 72. En otras palabras, el codificador de vídeo puede necesitar sintetizar la imagen de textura de referencia 72 con el fin de descodificar la imagen de textura actual 70. La imagen de textura de referencia 72 y la imagen de textura actual 70 están en la misma unidad de acceso, pero son vistas indiferentes.

Con el fin de sintetizar la imagen de textura de referencia 72, el codificador de vídeo puede procesar bloques (es decir, unidades de vídeo) de la imagen de textura actual 70. En el ejemplo de la FIG. 7, el codificador de vídeo está procesando un bloque actual 74. Cuando el codificador de vídeo procesa el bloque actual 74, el codificador de vídeo puede realizar el proceso NBDV para obtener un DV para el bloque actual 74. Por ejemplo, en el ejemplo de la FIG. 7, el codificador de vídeo identifica un DV 76 de un bloque 78 que está contiguo al bloque de vídeo actual 74. La identificación de DV 76 se muestra como Etapa 1 de la FIG. 7. Además, en el ejemplo de la FIG. 7, el codificador de vídeo determina, basándose en DV 76, un DV 78 del bloque actual 74. Por ejemplo, DV 78 puede ser una copia de DV 76. La copia de DV 76 se muestra como Etapa 2 de la FIG. 7.

El codificador de vídeo puede identificar, basándose en el DV 78 del bloque actual 74, un bloque de vídeo de referencia 80 en una imagen de profundidad de referencia 82. La imagen de profundidad de referencia 82, la imagen de textura actual 70 y la imagen de textura de referencia 72 pueden estar en la misma unidad de acceso. La imagen de profundidad de referencia 82 y la imagen de textura de referencia 72 pueden estar en la misma vista. El codificador de vídeo puede determinar, basándose en los valores de muestra de textura del bloque actual 74 y valores de muestra de profundidad del bloque de profundidad de referencia 80, valores de muestra de textura de la imagen de textura de referencia 72. El proceso de determinar los valores de la muestra de textura puede denominarse deformación hacia atrás. La deformación hacia atrás se muestra como el paso 3 de la FIG. 7. De esta manera, la FIG. 7 ilumina los tres pasos de cómo se localiza un bloque de profundidad desde la vista de referencia y luego se utiliza para la predicción BVSP.

El modo BVSP introducido se trata como un modo inter-codificado especial y se debe mantener un indicador de indicación del uso del modo BVSP para cada PU. En lugar de señalar el indicador en el flujo de bits, se agrega un nuevo candidato de fusión (candidato de fusión BVSP) para el modo de fusión a la lista de candidatos de fusión y el indicador depende de si el índice de candidato de fusión descodificado corresponde a un candidato de fusión BVSP. El candidato de fusión de BVSP se define de la forma siguiente:

1. Índice de imágenes de referencia para cada lista de imágenes de referencia: -1
2. Vector de movimiento para cada lista de imágenes de referencia: el vector de disparidad refinado

La posición insertada del candidato de fusión BVSP depende de los bloques espaciales contiguos:

1. Si alguno de los cinco bloques contiguos espaciales está codificado con el modo BVSP, es decir, el indicador mantenido del bloque contiguo es igual a 1, el codificador de vídeo trata al candidato de fusión BVSP como el candidato de fusión espacial correspondiente e inserta al candidato BVSP en la lista de candidatos de fusión. En algún ejemplo, el codificador de vídeo inserta el candidato de fusión BVSP en la lista de candidatos de fusión solo una vez.
2. De lo contrario (ninguno de los cinco bloques contiguos espaciales está codificado con el modo BVSP), el codificador de vídeo puede insertar el candidato de fusión BVSP en la lista de candidatos de fusión justo antes de los candidatos de fusión temporal.

En algunos ejemplos, durante el proceso de obtención de candidatos de fusión bi-predictivo combinado, un codificador de vídeo debería comprobar condiciones adicionales para evitar incluir el candidato de fusión BVSP.

Para cada PU codificada por BVSP con su tamaño denotado por $N \times M$, un codificador de vídeo puede dividir adicionalmente la PU en varias subregiones con el tamaño igual a $K \times K$ (donde K puede ser igual a 4). Para cada sub-región, el codificador de vídeo puede obtener un DMV separado, y cada sub-región puede predecirse desde un bloque localizado mediante el DMV derivado en la imagen de referencia de inter-vista. En otras palabras, el tamaño de una unidad de compensación de movimiento para las PU codificadas por BVSP puede ajustarse a $K \times K$. En condiciones de prueba comunes, K se fija en 4.

Para cada sub-región (bloque 4×4) dentro de una PU codificada con el modo BVSP, el codificador de vídeo puede localizar un bloque de profundidad de 4×4 correspondiente en la vista de profundidad de referencia con el DV refinado mencionado anteriormente. El codificador de vídeo puede seleccionar el valor máximo de los dieciséis píxeles de profundidad en el correspondiente bloque de profundidad. El codificador de vídeo puede convertir el valor

máximo en el componente horizontal de un DMV, y puede establecer que el componente vertical del DMV se establezca en 0.

La FIG. 8 es un diagrama conceptual que ilustra un ejemplo de estructura de predicción para las propuestas actuales de Predicción Residual Avanzada temporal (ARP) de un bloque de vídeo predicho temporalmente. ARP aplicado a las CU con el modo de partición igual a Part_2Nx2N fue adoptado en la 4.^a reunión de JCT3V, tal como se propone en Zhang et al., "CE4: Predicción Residual Avanzada para la codificación multivista", Equipo de Colaboración Conjunta en el Desarrollo de la Extensión de Codificación de Vídeo de la ITU-T SG 16 WP3 e ISO/IEC JTC 1/SC 29/WG 11, 4^a Reunión: Incheon, Corea, 20-26 de abril de 2013, Doc. JCT3V-D0177 (número MPEG m29008), en lo sucesivo "JCT3V-D0177". JCT3V-D0177 se puede descargar desde el siguiente enlace: http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=862.

Como se muestra en la FIG. 8, el codificador de vídeo invoca o identifica los siguientes bloques en la predicción del valor residual para un bloque de vídeo actual 100 en una imagen actual 102 de la vista actual, por ejemplo, dependiente V_m .

1. Bloque de vídeo actual 100 (en vista V_m): **Curr**

2. Un bloque de vídeo de referencia de inter-vista 106 en una imagen de referencia de inter-vista 108 de una vista de referencia / base (V_0 en la FIG 8): **Base**. El codificador de vídeo obtiene el bloque de vídeo de referencia de inter-vista 106 basándose en DV 104 del bloque de vídeo actual 100 (**Curr**). El codificador de vídeo puede determinar DV 104 usando NBDV, como se ha descrito anteriormente.

3. Un bloque de vídeo de referencia temporal 112 en una imagen de referencia temporal 114 **en la misma vista** (V_m) que el bloque de vídeo actual 100 (**Curr**): **CurrTRef**. El codificador de vídeo obtiene el bloque de vídeo de referencia temporal 112 basándose en el TMV110 del bloque de vídeo actual 100. El codificador de vídeo puede determinar TMV 100 usando cualquiera de las técnicas descritas en el presente documento.

4. Un bloque de vídeo de referencia temporal 116 en una imagen de referencia temporal 118 en la vista de referencia, es decir, la misma vista que el bloque de vídeo de referencia de inter-vista 106 (**Base**): **BaseTRef**. Un codificador de vídeo obtiene el bloque de vídeo de referencia temporal 116 en la vista de referencia usando TMV 110 del bloque de vídeo actual 100 (**Curr**). Un vector 120 de TMV+DV puede identificar el bloque de vídeo de referencia temporal 116 (**BaseTRef**) con respecto al bloque de vídeo actual 100 (**Curr**).

Cuando un codificador de vídeo predice temporalmente el bloque de vídeo actual 100 basándose en el bloque de vídeo de referencia temporal 112, que el codificador de vídeo identifica utilizando TMV 110, el codificador de vídeo determina las diferencias de píxel a píxel entre el bloque de vídeo actual 100 y el bloque de vídeo de referencia temporal 112 como un bloque residual. Estando ausente ARP, el codificador de vídeo transformaría, cuantificaría y codificaría por entropía el bloque residual. Un decodificador de vídeo decodificaría por entropía un flujo de bits de vídeo codificado, realizaría transformación y cuantificación inversa para obtener el bloque residual y aplicaría el bloque residual a una reconstrucción del bloque de vídeo de referencia 112 para reconstruir el bloque de vídeo actual 100.

Usando ARP, los codificadores de vídeo determinan un bloque de predicción residual que predice los valores del bloque residual, es decir, predice la diferencia entre el bloque de vídeo actual 100 (**Curr**) y el bloque de vídeo de referencia temporal 112 (**CurrTRef**). Un codificador de vídeo puede entonces solamente necesitar codificar una diferencia entre el bloque residual y el bloque de predicción residual, reduciendo la cantidad de información incluida en el flujo de bits de vídeo codificado para codificar el bloque de vídeo actual 100. En el ejemplo de ARP temporal de la FIG. 8, el predictor para el valor residual del bloque de vídeo actual 100 se determina basándose en bloques en la vista de referencia / base (V_0) que corresponden al bloque de vídeo actual 100 (**Curr**) y al bloque de vídeo de referencia temporal 112 (**CurrTRef**), y se identifican mediante DV 104. La diferencia entre estos bloques correspondientes en la vista de referencia puede ser un buen predictor del valor residual, es decir, la diferencia entre el bloque de vídeo actual 100 (**Curr**) y el bloque de vídeo de referencia temporal 112 (**CurrTRef**). En particular, los codificadores de vídeo identifican un bloque de vídeo de referencia de inter-vista 106 (**Base**) y un bloque de vídeo de referencia temporal 116 (**BaseTRef**) en la vista de referencia y determinan el bloque de predicción residual basándose en la diferencia entre el bloque de vídeo de referencia de inter-vista 106 y el bloque de vídeo de referencia temporal 116 (**Base-TRef-Base**), en el que la operación de sustracción se aplica a cada píxel de los conjuntos de píxeles denotados. En algunos ejemplos, un codificador de vídeo puede aplicar un factor de ponderación, w , al predictor residual. En tales ejemplos, el predictor final del bloque actual, es decir, el bloque de referencia sumado con el bloque de predicción residual, se puede designar como: **CurrTRef** + $w*(BaseTRef-Base)$.

La FIG. 9 es un diagrama conceptual que ilustra un ejemplo de estructura de predicción bidireccional para ARP temporal de un bloque de vídeo actual 120 en una vista actual (V_m). Las descripciones anteriores y la FIG. 8 ilustran la predicción unidireccional. Al extender ARP al caso de predicción bidireccional, el codificador de vídeo puede aplicar las técnicas anteriores a una o ambas listas de imágenes de referencia con el fin de identificar un bloque de predicción residual para el bloque de vídeo actual 120. En particular, el codificador de vídeo puede comprobar una o

ambas listas de referencia para el bloque de vídeo actual 100 para determinar si una de ellas contiene un TMV utilizable para ARP temporal. En el ejemplo ilustrado en la FIG. 9, el bloque de vídeo actual 120 está asociado con un TMV 130 que apunta a una primera imagen de referencia temporal 134 en una primera lista de imágenes de referencia (RefPicList0) y un TMV132 que apunta a una segunda imagen de referencia temporal 136 es una segunda lista de imágenes de referencia (RefPicList1).

En algunos ejemplos, un codificador de vídeo comprobará las listas de imágenes de referencia de acuerdo con un orden de comprobación para determinar si una de ellas incluye un TMV utilizable para ARP temporal y no necesita comprobar una segunda lista de acuerdo con la orden de comprobación si una primera lista incluye un TMV de este tipo. En algunos ejemplos, un codificador de vídeo comprobará ambas listas de imágenes de referencia y, si ambas listas incluyen un TMV, determinará qué TMV usar, por ejemplo, basándose en una comparación de los predictores residuales resultantes producidos usando los TMVs con respecto al valor residual del bloque de vídeo actual. En particular, de acuerdo con las propuestas actuales para ARP, cuando el bloque actual utiliza una imagen de referencia de inter-vista (en una vista diferente) para una lista de imágenes de referencia, el proceso de predicción residual está deshabilitado.

Como se ilustra en la FIG. 9, el codificador de vídeo puede utilizar un DV 124 identificado para el bloque de vídeo actual 120, por ejemplo, de acuerdo con NBDV, para identificar un bloque de vídeo de referencia de inter-vista 126 (*Base*) en una imagen de referencia de inter-vista 128 que está en una vista de referencia diferente (V_0), pero está en la misma unidad de acceso que la imagen actual 122. El codificador de vídeo puede utilizar también los TMV 130 y 132 para que el bloque de vídeo actual 120 identifique los bloques de referencia temporales (*BaseTRef*) para el bloque de vídeo de referencia de inter-vista 126 (*Base*) en varias imágenes de referencia temporales de la vista de referencia en ambas listas de imágenes de referencia, por ejemplo, RefPicList0 y RefPicList1. En el ejemplo de la FIG. 9, el codificador de vídeo identifica el bloque de vídeo de referencia temporal (*BaseTRef*) 140 en la imagen de referencia temporal 142 en una primera lista de imágenes de referencia, por ejemplo, RefPicList0 y el bloque de vídeo de referencia temporal (*BaseTRef*) 144 en la imagen de referencia temporal 146 en una segunda lista de imágenes de referencia, por ejemplo, RefPicList1, basándose en los TMV 130 y 132 del bloque de vídeo actual 120.

El uso de TMV 130 y 132 del bloque de vídeo actual 120 en la vista de referencia se ilustra mediante flechas discontinuas en la FIG. 9. En la FIG. 9, los bloques de vídeo de referencia temporales 140 y 144 en la vista de referencia se denominan bloques de referencia compensados por movimiento debido a su identificación basándose en los TMV 130 y 132. Un codificador de vídeo puede determinar un bloque de predicción residual para el bloque de vídeo actual 120 basándose en una diferencia entre el bloque de vídeo de referencia temporal 140 y el bloque de vídeo de referencia de inter-vista 126, o basándose en una diferencia entre el bloque de vídeo de referencia temporal 144 y el bloque de vídeo de referencia de inter-vista 126.

Se pueden describir los procedimientos principales de la ARP temporal propuesta en el lado del descodificador (con referencia a la FIG. 9) de la forma siguiente:

1. Un descodificador de vídeo obtiene un DV 124 como se especifica en el 3D-HEVC actual, por ejemplo, usando NBDV, apuntando a una vista de referencia objetivo (V_0). Entonces, en la imagen 128 de la vista de referencia dentro de la misma unidad de acceso, el descodificador de vídeo identifica el bloque de vídeo de referencia de inter-vista correspondiente 126 (*Base*) mediante DV 124.

2. El descodificador de vídeo vuelve a usar la información de movimiento, por ejemplo, TMV 130, 132, del bloque de vídeo actual 120 para obtener la información de movimiento para el correspondiente bloque de vídeo de referencia de inter-vista 126. El descodificador de vídeo puede aplicar una compensación de movimiento para el bloque de vídeo de referencia de inter-vista 126 basándose en TMV 130, 132 del bloque de vídeo actual 120 y la imagen de referencia obtenida 142, 146 en la vista de referencia para el bloque de vídeo de referencia 126, para identificar un bloque de vídeo de referencia temporal compensado por movimiento 140, 144 (*BaseTRef*) y determinar el bloque de predicción residual determinando *BaseTRef-Base*. La relación entre el bloque actual, el bloque correspondiente (*Base*) y el bloque compensado por movimiento (*BaseTRef*) se muestra en las FIGs. 8 y 9. En algunos ejemplos se selecciona como imagen de referencia del bloque correspondiente la imagen de referencia en la vista de referencia (V_0) que tiene el mismo valor POC (Recuento de Orden de Imágenes) que la imagen de referencia de la vista actual (V_m).

3. El descodificador de vídeo puede aplicar un factor de ponderación w al bloque de predicción residual para obtener un bloque de predicción residual ponderado y añadir los valores del bloque residual ponderado a las muestras predichas para reconstruir el bloque de vídeo actual 100.

En las propuestas de ARP, pueden utilizarse tres factores de ponderación, es decir, 0, 0,5 y 1. El que conlleva el coste de distorsión de la tasa mínima para la CU actual se selecciona como el factor de ponderación final y se transmite el índice del factor de ponderación correspondiente (0, 1 y 2 que corresponden al factor de ponderación 0, 1 y 0,5, respectivamente) en el flujo de bits al nivel de CU. Todas las predicciones de PU en una CU comparten el mismo factor de ponderación. Cuando el factor de ponderación es igual a 0, ARP no se utiliza para la CU actual.

Aspectos de ARP para 3D-HEVC se describen en Zhang et al., "3D-CE4: Predicción residual avanzada para la codificación multivista", Equipo de Colaboración Conjunta en el Desarrollo de la Extensión de Codificación de Vídeo de la ITU-T SG 16 WP 3 e ISO/IEC JTC1/SC 29/WG 11, 3.ª Reunión: Ginebra, Suiza, 17-23 de enero de 2013, Doc. JCT3V-C0049 (número MPEG m27784), en lo sucesivo "JCT3V-C0049". JCT3V-C0049 se puede descargar desde el siguiente enlace: http://phenix.int-evry.fr/jct3v/doc/end_user/current_document.php?id=487.

En JCT3V-C0049, las imágenes de referencia de diferentes PU codificadas con factores de ponderación distintos a cero pueden ser diferentes de PU a PU (o de bloque de vídeo actual a bloque de vídeo actual). Por lo tanto, es posible que se necesite acceder a diferentes imágenes desde la vista de referencia para generar el bloque compensado por movimiento (BaseTRef), por ejemplo, los bloques de vídeo de referencia temporales 116, 140 y 144 en las FIGs. 8 y 9 del correspondiente bloque de vídeo de referencia de inter-vista en la vista de referencia (Base), por ejemplo, los bloques de vídeo de referencia de inter-vista 106 y 126 en las FIGs. 8 y 9.

En JCT3V-D0177 se propuso una simplificación adicional de ARP a través de la selección de imágenes de referencia mediante escalado vectorial de movimiento. Por ejemplo, se propuso que un codificador de vídeo escalara los vectores de movimiento de la PU actual hacia una imagen fija antes de realizar la compensación de movimiento para el proceso de generación residual cuando el factor de ponderación es diferente a 0. En JCT3V-D0177, la imagen fija se define como la primera imagen de referencia de cada lista de imágenes de referencia si es desde la misma vista. Cuando el vector de movimiento descodificado no apunta a la imagen fija, primero es escalado por el codificador de vídeo, y luego utilizado por el codificador de vídeo para identificar CurrTRef y BaseTRef para el bloque de vídeo actual. Dicha imagen de referencia utilizada para ARP puede denominarse imagen de referencia de ARP objetivo. Pueden existir dos imágenes de referencia de ARP objetivo, correspondientes a RefPicList0 y RefPicList1, respectivamente, que se pueden designar como la imagen de referencia de ARP objetivo L0 y la imagen de referencia de ARP objetivo L1, respectivamente.

De acuerdo con JCT3V-C0049, un codificador de vídeo aplica un filtro bi-lineal durante un proceso de interpolación del bloque correspondiente (Base) y su bloque de predicción (BaseTRef), pero aplica un filtro convencional de 8/4 coeficientes para un proceso de interpolación del bloque de vídeo actual (Curr), por ejemplo, PU, y el bloque de predicción (CurrTRef) del bloque de vídeo actual. Se propuso en JCT3V-D0177 que el codificador de vídeo emplee siempre filtros bilineales para dichos procesos de interpolación independientemente de si el bloque está en vista de base o no de base cuando se aplica ARP.

Además, de acuerdo con las propuestas existentes para ARP, la vista de referencia para ARP se identifica mediante el índice de orden de vista devuelto desde el proceso NBDV. Como se describió anteriormente, un codificador de vídeo puede utilizar el proceso NBDV para determinar el DV, por ejemplo, DV 104 o 124, utilizado para identificar el correspondiente bloque de vídeo de referencia de inter-vista correspondiente (base), por ejemplo, los bloques de vídeo de referencia de inter-vista 106 y 126 en las FIGs. 8 y 9. De acuerdo con las propuestas existentes para ARP, cuando la imagen de referencia de un bloque de vídeo (PU) en una lista de imágenes de referencia es de una vista diferente, entonces la vista de referencia objetivo para ARP, identificada por el índice de orden de vista devuelto desde el proceso NBDV, está deshabilitada para esta lista de imágenes de referencia.

Puede haber problemas asociados con las propuestas existentes para ARP en 3D-HEVC. Por ejemplo, de acuerdo con las propuestas existentes, ARP solo predice el valor residual generado a partir de la predicción temporal, cuando el vector de movimiento actual de un bloque de vídeo actual se refiere a una imagen de referencia en la misma vista. Por lo tanto, cuando el vector de movimiento actual de un bloque de vídeo actual se refiere a una imagen de referencia de inter-vista, ARP no es aplicable, aunque el (los) elemento(s) sintáctico(s) relacionado(s) con ARP signe(n) siendo transmitido(s).

Como otro ejemplo, un DV obtenido, por ejemplo, según se obtiene de acuerdo con un proceso de NBDV, puede ser menos preciso en comparación con un DMV explícito, que se elige típicamente mediante optimización de distorsión de velocidad (RDO). Además, como proceso de descodificación, la predicción de movimiento, incluida la predicción de inter-vista, ocurre después de la generación de DV, y la ARP ocurre después de la predicción de movimiento. Por lo tanto, los TMV o DMVs más precisos que podrían usarse para identificar diferentes bloques que no se consideran en ARP actual están disponibles cuando ARP es realizado por un codificador de vídeo. Sin embargo, como se ha descrito anteriormente, con referencia a las FIGs. 8 y 9, las propuestas existentes para ARP temporal utilizan el DV obtenido a través de NBDV para identificar el bloque de vídeo de referencia de inter-vista correspondiente.

Esta divulgación proporciona técnicas que pueden abordar problemas asociados con las propuestas existentes para ARP, incluyendo aquellas analizadas anteriormente, y pueden así mejorar la eficacia de codificación de ARP. Por ejemplo, un codificador de vídeo, por ejemplo, un codificador de vídeo 20 y/o un descodificador de vídeo 30, que implementa las técnicas de esta divulgación para codificar un bloque de vídeo actual utilizando ARP, puede identificar un DMV desde una vista actual del bloque de vídeo actual a una vista de referencia y determinar un bloque de predicción residual para el bloque de vídeo actual basándose en la identificación del DMV. En algunos ejemplos, el DMV es un DMV usado para la predicción de inter-vista del bloque de vídeo actual, y el codificador de vídeo puede realizar ARP de inter-vista para codificar el bloque de vídeo actual. En otros ejemplos, el DMV es un DMV de un bloque de vídeo de referencia temporal en la misma vista que el bloque de vídeo actual. En tales

ejemplos, el DMV puede usarse en ARP temporal para el bloque de vídeo actual en lugar del DV obtenido mediante NBDV para el bloque de vídeo actual.

La FIG. 10 es un diagrama conceptual que ilustra un ejemplo de estructura de predicción para ARP de inter-vista de un bloque de vídeo predicho de inter-vista de acuerdo con las técnicas descritas en esta divulgación. De acuerdo con la técnica de ejemplo ilustrada en la FIG. 10, un codificador de vídeo, p. ej., codificador de vídeo 20 y/o descodificador de vídeo 30, puede utilizar un valor residual de inter-vista calculado en una unidad de acceso diferente para predecir el valor residual del bloque actual, que se predice mediante inter-vista. En contraste con las propuestas de ARP en las que ARP no se realiza cuando el vector de movimiento del bloque actual es un DMV, y se realiza solamente cuando el vector de movimiento para el bloque de vídeo actual es un TMV, la técnica de ejemplo de la FIG. 10 utiliza el DMV para realizar ARP.

En particular, la técnica a modo de ejemplo de la FIG. 10 puede realizarse mediante un codificador de vídeo, por ejemplo, un codificador de vídeo 20 o un descodificador de vídeo 30, cuando el vector de movimiento del bloque de vídeo actual 150 (Curr) en una imagen actual 152 es un DMV 154 y el bloque de vídeo de referencia de inter-vista 156 (Base) en una imagen de referencia de inter-vista 158 en la vista de referencia (V_0) contiene al menos un TMV 160. En algunos ejemplos, el DMV 154 puede ser un DV que se convirtió a un DMV para actuar como un IDMVC para la predicción de información de movimiento del bloque de vídeo actual 150.

El codificador de vídeo identifica el bloque de vídeo de referencia de inter-vista 156 (**Base**) en la imagen de referencia de inter-vista 158 usando el DMV 154 para el bloque de vídeo actual 150. El codificador de vídeo utiliza un TMV 160 y una imagen de referencia asociada, por ejemplo, una imagen de referencia temporal 164 en la vista de referencia (V_0), del bloque de vídeo de referencia de inter-vista 156 junto con el DMV para identificar un bloque de vídeo de referencia temporal 162 (**BaseTRef**) en la imagen de referencia temporal 164 en la vista de referencia (V_0). La identificación del bloque de vídeo de referencia temporal 162 (**BaseTRef**) basándose en TMV 160 y DMV 154 está representada por el vector de trazos 170 (TMV + DMV). El codificador de vídeo también usa TMV 160 para identificar un bloque de referencia temporal 166 (**CurrTRef**) en una imagen de referencia temporal 168 en la vista actual (V_m). El bloque de vídeo de referencia temporal 162 (**BaseTRef**) en la vista de referencia (V_0) y el bloque de vídeo de referencia temporal 166 (**CurrTRef**) en la vista actual (V_m) puede estar dentro de la misma unidad de acceso, es decir, la imagen de referencia temporal 164 en la vista de referencia (V_0) y la imagen de referencia temporal 168 en la vista actual (V_m) pueden estar en la misma unidad de acceso.

El codificador de vídeo, por ejemplo, el codificador de vídeo 20 y/o el descodificador de vídeo 30, pueden entonces calcular el bloque de predicción residual de inter-vista en una unidad de acceso diferente del bloque de vídeo actual 150 basándose en la diferencia de píxel a píxel entre estos dos últimos bloques, es decir, la diferencia entre el bloque de vídeo de referencia temporal 166 en la vista actual y el bloque de vídeo de referencia temporal 164 en la vista de referencia, o **CurrTRef-BaseTRef**. La señal de diferencia, denominada predictor residual de inter-vista, puede usarse para predecir el valor residual del bloque de vídeo actual 150. La señal de predicción del bloque de vídeo actual 150 puede ser la suma del predictor de inter-vista, es decir, el bloque de vídeo de referencia de inter-vista 156 (**Base**) y el valor residual de inter-vista predicho en una unidad de acceso diferente determinada basándose en la diferencia entre el bloque de vídeo de referencia temporal 166 en la vista actual y el bloque de vídeo de referencia temporal 164 en la vista de referencia. En algunos ejemplos, se aplica un factor de ponderación w al valor residual de inter-vista predicho en la unidad de acceso diferente. En tales ejemplos, la señal de predicción del bloque de vídeo actual 150 puede ser: **Base + w *(CurrTRef-BaseTRef)**.

En algunos ejemplos, el codificador de vídeo puede determinar una imagen de referencia objetivo en una unidad de acceso objetivo para ARP de inter-vista, por ejemplo, similar a la determinación de una imagen de referencia objetivo para ARP temporal, como se analizó anteriormente. En algunos ejemplos, como se ha analizado anteriormente con referencia a JCT3V-D0177, la imagen de referencia objetivo para cada lista de imágenes de referencia es la primera imagen de referencia en la lista de imágenes de referencia. En otros ejemplos, la imagen de referencia objetivo, por ejemplo, POC objetivo, para una o ambas listas de imágenes de referencia puede ser señalada desde el codificador de vídeo 20 al descodificador de vídeo 30, por ejemplo, en una PU, CU, fragmento, imagen u otra base. En otros ejemplos, la imagen de referencia objetivo para cada lista de imágenes de referencia es la imagen de referencia temporal en la lista de imágenes de referencia que tiene la diferencia de POC más pequeña comparada con el bloque actual y el índice de imagen de referencia más pequeño. En otros ejemplos, la imagen de referencia objetivo para dos listas de imágenes de referencia es la misma.

Si la imagen que contiene el bloque de vídeo de referencia temporal en la vista de referencia indicada por TMV 160 está en una unidad de acceso diferente (instancia de tiempo) a la imagen de referencia de ARP objetivo, el codificador de vídeo puede escalar TMV 160 a la imagen de referencia objetivo, por ejemplo, imagen de referencia objetivo 164, para identificar el bloque de vídeo de referencia temporal 162 (**BaseTRef**) en la vista de referencia para el ARP de inter-vista. En tales ejemplos, el codificador de vídeo localiza el bloque de vídeo de referencia temporal 162 en la unidad de acceso que contiene la imagen de referencia de ARP objetivo. El codificador de vídeo puede escalar TMV 160 mediante escalado de POC. Además, el TMV escalado se utiliza para identificar el bloque de vídeo de referencia temporal (**CurrTRef**) 166 en la vista actual que se encuentra en la imagen de referencia de ARP objetivo.

En algunos ejemplos, el codificador de vídeo escala TMV 160 a la imagen de referencia objetivo LX (X siendo 0 o 1), donde LX corresponde a la RefPicListX de la PU que incluye el TMV. En algunos ejemplos, el codificador de vídeo puede escalar TMVs desde uno o ambos de RefPicList0 o RefPicList1 a la imagen de referencia objetivo L0 o L1, respectivamente. En algunos ejemplos, el codificador de vídeo escala TMV 160 a la imagen de referencia objetivo LX, en la que X satisface la condición de que el DMV 154 del bloque de vídeo actual 150, por ejemplo, la PU actual, corresponde a RefPicListX.

De forma similar, en algunos ejemplos, el codificador de vídeo escala DMV 154 a una vista de referencia objetivo para ARP antes de identificar el bloque de vídeo de referencia de inter-vista 156 en la imagen de referencia 158 en la vista de referencia objetivo. El codificador de vídeo puede escalar el DMV 154 mediante escalado de diferencia de orden de vista. La vista de referencia objetivo puede ser predeterminada y conocida por el codificador de vídeo 20 y el decodificador de vídeo 30, o puede ser señalada desde el codificador de vídeo 20 al decodificador de vídeo 30, por ejemplo, en una PU, CU, fragmento, imagen u otra base.

En algunos ejemplos de ARP de inter-vista, el codificador de vídeo, por ejemplo, el codificador de vídeo 20 y/o el decodificador de vídeo 30, pueden obtener la señal de predicción para el bloque actual 150 usando la misma estructura de predicción y los bloques de vídeo de referencia 156, 164 y 168 identificados ilustrados en la FIG. 10, pero determina el bloque de predicción residual basándose en la diferencia entre los bloques de referencia 156 y 162 en la vista de referencia, en lugar de los bloques de referencia 162 y 166 en la unidad de acceso diferente. En tales ejemplos, el codificador de vídeo puede aplicar un factor de ponderación a otras matrices de muestra, por ejemplo, la diferencia entre los bloques de referencia 156 y 162 en la vista de referencia y, en consecuencia, obtener la señal de predicción para el bloque de vídeo actual 150 de la forma siguiente: $\text{CurrTRef} + w * (\text{Base} - \text{BaseTRef})$. En algunos ejemplos de ARP de inter-vista, el codificador de vídeo puede utilizar varios filtros de interpolación, incluyendo filtros bilineales, para obtener los bloques de vídeo de referencia 156, 162 y 166 si están alineados con posiciones de píxeles fraccionadas.

Aunque la FIG. 10 ilustra ejemplos de ARP de inter-vista en los que se identifican los bloques de vídeo de referencia temporales en las vistas actuales y de referencia utilizando el TMV y la imagen de referencia asociada del bloque de vídeo de referencia de inter-vista, en otros ejemplos pueden usarse otras TMVs e imágenes de referencia asociadas para identificar los bloques de vídeo de referencia temporales en las vistas actual y de referencia. Por ejemplo, si el DMV del bloque de vídeo actual es de una primera lista de imágenes de referencia (por ejemplo, RefPicList0 o RefPicList1) del bloque de vídeo actual, un codificador de vídeo puede usar un TMV y una imagen de referencia asociada de una segunda lista de imágenes de referencia (por ejemplo, la otra de RefPicList0 o RefPicList1) del bloque de vídeo actual. En dichos ejemplos, el codificador de vídeo puede identificar el bloque de vídeo de referencia temporal en la vista actual en la imagen de referencia asociada con el TMV, o escalar el TMV a una unidad de acceso objetivo y una imagen de referencia objetivo para ARP para identificar el bloque de vídeo de referencia temporal en la vista actual. En tales ejemplos, el codificador de vídeo puede identificar el bloque de vídeo de referencia temporal en la vista de referencia en una imagen de referencia que está en la misma unidad de acceso que la imagen de referencia en la que se localizó el bloque de vídeo de referencia temporal en la vista actual. En otros ejemplos, en lugar del TMV del bloque de vídeo de referencia de inter-vista o el TMV de la otra lista de imágenes de referencia del bloque de vídeo actual, un codificador de vídeo puede utilizar de manera similar un TMV y una imagen de referencia asociada derivada de la información de movimiento de los bloques de vídeo contiguos espaciales o temporales del bloque de vídeo actual para identificar los bloques de vídeo de referencia temporales en las vistas actuales y de referencia para ARP.

La FIG. 10 ilustra un ejemplo de ARP de inter-vista de acuerdo con la presente divulgación. Como se ha analizado anteriormente, de acuerdo con las propuestas existentes de ARP temporal, se utiliza un DV del bloque de vídeo actual, por ejemplo, obtenido mediante NBDV, para identificar el bloque de vídeo de referencia de inter-vista en la vista de referencia. De acuerdo con las técnicas de esta divulgación, la precisión del predictor residual temporal calculado en una vista de referencia para ARP temporal puede aumentarse reemplazando el DV por un DMV del bloque de referencia temporal en la vista actual (**CurrTRef**) si contiene al menos un DMV.

La FIG. 11 es un diagrama conceptual de un ejemplo de estructura de predicción para la ARP temporal de un bloque de vídeo actual predicho temporalmente 180 en una imagen actual 182 que utiliza un DMV 190 del bloque de referencia temporal en la vista actual (**CurrTRef**) de acuerdo con las técnicas descritas en esta divulgación. De acuerdo con el ejemplo de la FIG. 11, un codificador de vídeo, por ejemplo, el codificador de vídeo 20 y/o el decodificador de vídeo 30 predice temporalmente el bloque de vídeo actual 180 usando un TMV 184 que identifica un bloque de vídeo de referencia temporal 186 en una imagen de referencia temporal 188. El codificador de vídeo determina si el bloque de vídeo de referencia temporal 186 contiene al menos un DMV, por ejemplo DMV 190, usado para el bloque de vídeo de referencia temporal de predicción de inter-vista 186. En algunos ejemplos, el DMV 190 puede ser un IDMVC utilizado para la predicción de información de movimiento del bloque de vídeo de referencia temporal 186.

El codificador de vídeo puede utilizar DMV 190 en lugar de un DV del bloque de vídeo actual 180 para la identificación de uno o ambos de: un bloque de vídeo de referencia de inter-vista 196 (Base) dentro de una imagen

de referencia 198 en la vista de referencia (V0), o un bloque de vídeo de referencia temporal 194 (BaseTRef) en la imagen de referencia temporal 194 en la vista de referencia (V0). La identificación del bloque de vídeo de referencia temporal 194 basándose en TMV 184 y DMV 190 se ilustra mediante el vector 200, que está marcada como TMV + DMV. En algunos ejemplos, cuando el codificador de vídeo utiliza un DMV para reemplazar el DV de NBDV para ARP temporal, el codificador de vídeo también puede reemplazar el índice de orden de vista devuelto del proceso NBDV por un índice de orden de vista asociado con el DMV seleccionado. Adicionalmente, en algunos ejemplos, un codificador de vídeo tal vez no seleccione un DMV asociado con el bloque de vídeo de referencia temporal 186 para ARP temporal del bloque de vídeo actual 180 si el DMV se obtuvo usando el modo BVSP para reemplazar el DV de NBDV. El codificador de vídeo puede determinar el bloque de predicción residual temporal para el bloque de vídeo actual 180 usando los bloques de vídeo de referencia identificados 186, 192 y 196, como se ha descrito anteriormente con referencia a los bloques 106, 112 y 116 de la FIG. 8.

En algunos ejemplos, si el TMV 184 descodificado del bloque de vídeo actual 180 apunta a una imagen de referencia en una unidad de acceso diferente (instancia de tiempo) a la imagen de referencia de ARP objetivo, el codificador de vídeo puede escalar TMV 184 a la imagen de referencia de ARP objetivo 188 y localizar el bloque de vídeo de referencia temporal 186 (**CurrTRef**) en la imagen de referencia de ARP objetivo mediante TMV 184 escalado, por ejemplo, usando escalado de POC. En tales ejemplos, el codificador de vídeo puede obtener el DMV 190 en el bloque de vídeo de referencia temporal 186 (**CurrTRef**) como se identifica mediante TMV 184 escalado. En algunos ejemplos, cuando el codificador de vídeo escala TMV 184 para identificar el bloque de vídeo de referencia temporal 186 (**CurrTRef**) en la imagen 188 perteneciente a la misma unidad de acceso que la imagen ARP objetivo, puede identificarse otro bloque de vídeo de referencia temporal identificado por TMV 184 sin escalar, es decir, **CurrTempRef**. En tales ejemplos, el codificador de vídeo puede usar un DMV de este bloque de vídeo de referencia temporal (**CurrTempRef**), si está disponible, para reemplazar el DV por ARP temporal del bloque de vídeo actual 180. En algunos ejemplos, el codificador de vídeo solo identifica y usa **CurrTempRef** cuando no hay DMV asociado con el bloque de vídeo de referencia temporal 186 (**CurrTRef**). En algunos ejemplos, se pueden usar otros DMV de bloques codificados para reemplazar el DV de NBDV.

La FIG. 12 es un diagrama conceptual que ilustra técnicas de ejemplo para la identificación de un TMV o DMV en o adyacente a un bloque de vídeo de acuerdo con las técnicas descritas en esta divulgación. Como se ha analizado anteriormente, con respecto a las FIGs. 10 y 11, un codificador de vídeo, por ejemplo, un codificador de vídeo 20 y/o un descodificador de vídeo 30, identifica TMVs y DMV para implementar ARP de inter-vista y ARP temporal de acuerdo con las técnicas descritas en esta divulgación. En algunos ejemplos, el codificador de vídeo identifica los TMV y los DMV en o adyacentes a un bloque de vídeo actual, o un bloque de vídeo de referencia temporal o de inter-vista, que puede ser una región dentro de una imagen de referencia con el mismo tamaño que el bloque de vídeo actual, por ejemplo, PU actual, de anchura $\times$ altura.

La FIG. 12 ilustra un bloque 210 de anchura $\times$ altura. El bloque 210 puede ser un bloque de vídeo actual o un bloque de vídeo de referencia temporal o de inter-vista que puede ser una región dentro de una imagen de referencia con el mismo tamaño que el bloque de vídeo actual. La FIG. 12 ilustra también un bloque 212 adyacente o que incluye la posición central del bloque 210 y un bloque 214 adyacente o que incluye una posición inferior derecha del bloque 210.

En algunos ejemplos, para ARP temporal o de inter-vista, el codificador de vídeo considera, por ejemplo, solo el vector de movimiento, por ejemplo, TMV o DMV y el índice de referencia asociado con una PU u otro bloque que contiene la posición central del bloque, por ejemplo, el bloque 212 dentro del bloque 210. En algunos ejemplos, el codificador de vídeo considera, por ejemplo, solo la información de movimiento (incluyendo los vectores de movimiento y los índices de referencia) de dos bloques que contienen los píxeles de la parte inferior derecha (con coordinación con respecto al píxel de la esquina superior izquierda de (anchura, altura)), y central (con una coordinación relativa a la esquina superior izquierda de (anchura/2, altura/2)) del bloque 210. Con referencia a la FIG. 12, los bloques 214 y 212, respectivamente, pueden ser ejemplos de bloques que contienen los píxeles de la parte inferior derecha y central del bloque 210. Los bloques 212 y 214 pueden ser $N \times N$, que puede ser la granularidad de bloque más grande que puede contener hasta un vector de movimiento correspondiente a cada lista de imágenes de referencia; por ejemplo, $N \times N$ puede ser 4×4 . Los bloques 212 y 214 se pueden comprobar para el DMV o TMV para ARP en cualquier orden.

En algunos ejemplos, suponiendo que el bloque de vídeo actual, por ejemplo, el PU actual, tiene una coordenada de (x, y) y el vector utilizado para identificar el bloque de vídeo de referencia ($v[0]$, $v[1]$) es de un vector de movimiento (TMV o DMV), el codificador de vídeo puede convertir el DMV como $v[i] = (mv[i] + 2) \gg 2$, para i igual a 0 o 1 respectivamente, o $v[i] = mv[i] \gg 2$. En tales ejemplos, el codificador de vídeo puede identificar los bloques 212 y 214 respectivamente como el bloque, por ejemplo, bloque 4×4 , cubriendo el píxel con coordinación de $(x+v[0]+anchura/2, y+v[1]+altura/2)$ y el bloque que cubre el píxel con coordinación $(x+v[0]+anchura, y+v[1]+altura)$. En algunos ejemplos, el codificador de vídeo puede identificar uno o ambos del bloque central 212 y del bloque inferior derecho 214 por coordinación desplazada con $(-1, -1)$, que corresponde a $(x+v[0]+anchura/2-1, y+v[1]+altura/2-1)$ y $(x+v[0]+anchura-1, y+v[1]+altura-1)$ respectivamente.

En algunos ejemplos, el codificador de vídeo puede comprobar los bloques 212 y 214 para un TMV o DMV

disponible de acuerdo con una orden de comprobación. En algunos ejemplos, el codificador de vídeo puede comprobar primero el bloque central 212 y utilizar el DMV o TMV asociado con el bloque central para ARP, si dicho vector de movimiento está disponible. En tales ejemplos, el codificador de vídeo puede comprobar el bloque inferior derecho 214 para un TMV o DMV para ARP si dicho vector de movimiento no estaba disponible desde el bloque central 212.

En algunos ejemplos, el codificador de vídeo puede comprobar las listas de imágenes de referencia para los bloques 212, 214 para un vector de movimiento apropiado para ARP en un orden de comprobación. Por ejemplo, el codificador de vídeo puede comprobar RefPicList0, y utilizar el DMV o TMV asociado con RefPicList0, si dicho vector de movimiento está disponible. En tales ejemplos, el codificador de vídeo puede comprobar RefPicList1 para un TMV o DMV para ARP si dicho vector de movimiento no estaba disponible en RefPicList0.

En algunos ejemplos, el codificador de vídeo puede considerar, por ejemplo, solo considerar, los vectores de movimiento asociados con las PU que contienen la posición central y una o más de las cuatro posiciones de esquina del bloque. El codificador de vídeo puede considerar las PU en un orden basándose en una prioridad y, una vez que se encuentre un vector de movimiento, puede no considerar las otras PU. La prioridad de estas diferentes posiciones puede, en un ejemplo, definirse como: central, superior izquierda, superior derecha, inferior izquierda e inferior derecha del bloque.

En algunos ejemplos, el codificador de vídeo puede configurarse para considerar toda la información de movimiento asociada con el bloque. En algunos ejemplos, el codificador de vídeo tal vez no compruebe los vectores de movimiento adicionales una vez que se encuentre un TMV o DMV para ARP dentro del bloque 210. Una prioridad para comprobar PU dentro del bloque 210 para vectores de movimiento puede ser, por ejemplo, un orden de escaneado de trama o escaneado en espiral. Algunos ejemplos de órdenes de escaneado en espiral para escanear un bloque, por ejemplo, un bloque 4x4, para vectores de movimiento se representan en las FIGs. 13A - 13D.

En algunos ejemplos, al comprobar un bloque 210 para un TMV para ARP, un codificador de vídeo tal vez solo considere TMVs apuntando a una imagen de referencia en la misma unidad de acceso que la imagen de referencia de ARP objetivo. En algunos ejemplos, al comprobar un bloque de referencia 210 para un DMV para ARP, un codificador de vídeo tal vez solo considere DMVs que apuntan a una imagen de referencia de inter-vista en una misma vista que la indicada por el DMV o DV para el bloque de vídeo actual. En algunos ejemplos, el codificador de vídeo extiende primero el bloque 210 a un bloque asociado con una PU y busca un TMV o un DMV dentro del bloque extendido para ser el TMV o DMV para ARP. En algunos ejemplos, si no se encuentra TMV o DMV en el bloque 210, el codificador de vídeo realiza ARP usando un vector de movimiento cero, o no realiza ARP. En algunos ejemplos, cuando un codificador de vídeo utiliza un vector de movimiento cero para identificar los dos bloques de vídeo de referencia temporales en la vista actual y de referencia, el codificador de vídeo puede usar la imagen de referencia objetivo para RefPicListX, donde X puede ser 0 o 1 e indicar qué lista fue invocada para la predicción de inter-vista, por ejemplo, qué lista incluía el DMV.

Como se ha descrito anteriormente, por ejemplo, con referencia a la FIG. 12, un codificador de vídeo puede identificar un TMV o un DMV en un bloque dado que contiene solamente hasta dos conjuntos de información de movimiento, por ejemplo, un bloque 4x4 dentro del bloque 210. Un conjunto de información de movimiento corresponde a una primera lista de imágenes de referencia, por ejemplo, la Lista de Imágenes de Referencia 0 (RefPicList0) para el bloque dado y la otra corresponde a una segunda lista de imágenes de referencia, por ejemplo, la Lista de Imágenes de Referencia 1 (RefPicList1), para el bloque dado. Cada conjunto de información de movimiento incluye un vector de movimiento y un índice de referencia.

En algunos ejemplos, un codificador de vídeo solo considera la información de movimiento correspondiente a RefPicList0 para identificar un TMV o DMV para ARP de un bloque de vídeo actual. En otros ejemplos, un codificador de vídeo solo considera la información de movimiento correspondiente a RefPicList1 para identificar un TMV o DMV para ARP de un bloque de vídeo actual. En otros ejemplos, el codificador de vídeo considera la información de movimiento correspondiente a RefPicListX primero. Si la información de movimiento correspondiente a RefPicListX no incluye un TMV o DMV adecuado para ARP, el codificador de vídeo considera la información de movimiento correspondiente a RefPicListY (con Y igual a 1-X).

En algunos ejemplos, X es igual a 0. En algunos ejemplos, X es igual a 1. En algunos ejemplos, X es igual a Z, en el que Z corresponde a la lista de imágenes de referencia en la que se incluyó el vector de movimiento (TMV o DMV) del bloque de vídeo actual. Por ejemplo, si el vector de movimiento perteneciente al bloque de vídeo actual, por ejemplo, PU actual, corresponde a RefPicList0, Z es 0. Si el vector de movimiento perteneciente al bloque de vídeo actual, por ejemplo, la PU actual, corresponde a RefPicList1, Z es 1. En algunos ejemplos, el codificador de vídeo solo codifica la información de movimiento correspondiente a RefPicListZ.

Otro ejemplo de técnica para mejorar la precisión de la ARP temporal, por ejemplo, la exactitud del valor residual temporal calculado en una vista de referencia, reemplazando el DV del bloque de vídeo actual, incluye reemplazar el DV, por ejemplo, obtenido mediante NBDV, por un DV obtenido de un bloque de profundidad co-ubicado de un bloque de referencia temporal del bloque actual (**CurrTRef**). Un codificador de vídeo, por ejemplo, el codificador de

vídeo 20 y/o decodificador de vídeo 30, puede obtener el DV a través de un bloque de profundidad co-ubicado de un bloque de referencia temporal del bloque actual (**CurrTRef**) usando técnicas similares o iguales a las utilizadas para obtener un DV para un bloque de vídeo actual para BVSP, como se ha descrito anteriormente con respecto a la FIG. 7.

El codificador de vídeo puede utilizar el DV obtenido a través del bloque de profundidad co-ubicado del bloque de referencia temporal del bloque actual (**CurrTRef**) en lugar del DV del actual, por ejemplo, obtenido mediante NBDV, para la obtención de cualquiera de los dos o los dos bloques de referencia en la vista de referencia utilizada para ARP. Por ejemplo, el codificador de vídeo puede utilizar el DV obtenido a través del bloque de profundidad co-ubicado del bloque de referencia temporal del bloque actual (**CurrTRef**) para identificar uno o ambos bloques de referencia de inter-vista del bloque actual en la vista de referencia (**Base**) o el bloque de referencia temporal en la vista de referencia (**BaseTRef**). El codificador de vídeo puede identificar el bloque de referencia temporal en la vista de referencia (**BaseTRef**) añadiendo el TMV del bloque actual al DV obtenido a través del bloque de profundidad co-ubicado del bloque de referencia temporal del bloque actual (**CurrTRef**).

Como se ha analizado anteriormente, en algunos ejemplos, si el TMV descodificado del bloque actual apunta a una imagen de referencia en una unidad de acceso diferente (instancia de tiempo) a la imagen de referencia de ARP objetivo, el codificador de vídeo puede escalar el TMV a la imagen de referencia de ARP objetivo, y localizar **CurrTRef** mediante el TMV escalado. En tales ejemplos, el codificador de vídeo obtiene el DV del bloque de profundidad co-ubicado del bloque de referencia temporal del bloque actual (**CurrTRef**) como se identifica mediante el TMV escalado. Además, como se ha analizado anteriormente, en algunos ejemplos, cuando se escala un TMV para identificar el **CurrTRef** en una imagen que pertenece a la misma unidad de acceso que la imagen ARP objetivo, el codificador de vídeo puede identificar otro bloque de referencia temporal identificado por el TMV sin escalar, es decir, se puede identificar un **CurrTempRef**, y un DV derivado del bloque de profundidad co-ubicado de **CurrTempRef**, si está disponible, puede ser utilizado para reemplazar el DV. En algunos ejemplos, el codificador de vídeo solo necesita identificar y utilizar **CurrTempRef** cuando no puede obtener un mediante un bloque de profundidad co-ubicado de un bloque de referencia temporal del bloque actual (**CurrTRef**).

El codificador de vídeo, por ejemplo, el codificador de vídeo 20 y/o el decodificador de vídeo 30, pueden obtener un DV del bloque de profundidad co-ubicado del bloque de referencia temporal del bloque actual (**CurrTRef**) en cualquiera de una variedad de formas. En algunos ejemplos, el codificador de vídeo utiliza solo una muestra dentro del bloque de profundidad co-ubicado directamente, y convierte el valor de profundidad asociado en el DV para ARP temporal. En algunos ejemplos, la muestra individual del bloque de profundidad co-ubicado utilizado para obtener el DV para ARP temporal es el píxel situado en el centro del bloque de profundidad co-ubicado, por ejemplo, en (W/2, H/2) relativo a la muestra superior izquierda de un bloque de profundidad con un tamaño de WxH.

En algunos ejemplos, el codificador de vídeo utiliza varias muestras selectivas dentro del bloque de profundidad co-ubicado para determinar un valor de profundidad representativo, por ejemplo, a través de una función matemática. En un ejemplo, el codificador de vídeo selecciona cuatro muestras de profundidad de esquina. En otro ejemplo, el codificador de vídeo selecciona las muestras de profundidad dentro del bloque de profundidad co-ubicado basándose en las muestras de profundidad contiguas del bloque de profundidad. Por ejemplo, cuando las muestras de profundidad contiguas muestran un borde horizontal, el codificador de vídeo puede seleccionar solamente los dos píxeles de esquina en la primera fila. En algunos ejemplos, todas las muestras de profundidad dentro del bloque de profundidad co-ubicado pueden usarse para determinar un valor de profundidad representativo a través de una función matemática. En algunos ejemplos, el codificador de vídeo puede determinar un valor de profundidad representativo basándose en valores de profundidad seleccionados (o todos) a partir del bloque de profundidad co-ubicado, por ejemplo, determinando un valor máximo, promedio o mediana de los valores de profundidad seleccionados o aplicando alguna otra función a los valores de profundidad seleccionados.

En algunos ejemplos, un codificador de vídeo puede aplicar las técnicas ARP temporales descritas anteriormente que implican el DV obtenido a través del bloque de profundidad co-ubicado del bloque de referencia temporal del bloque actual (**CurrTRef**) cuando no es necesario descodificar una vista de textura independiente de la vista de profundidad asociada. Cuando se requiere la descodificación de una vista de textura independiente de la vista de profundidad asociada, un codificador de vídeo puede aplicar otras técnicas de ARP descritas en el presente documento, tales como las descritas con respecto a las FIGs. 10 y 11.

Cuando tanto la ARP temporal como de inter-vista están habilitadas, la condición de señalización del factor de ponderación para ARP puede cambiarse de comprobar si todas las imágenes de referencia son imágenes de referencia de inter-vista a simplemente comprobar si la imagen actual es una imagen de acceso aleatorio (IRAP, con un tipo de unidad NAL de 15 a 22, incluidos: es decir, BLA_W_LP, BLA_W_RADL, BLA_N_LP, IDR_W_RADL, IDR_N_LP o CRA_NUT). Por lo tanto, en algunos ejemplos, un codificador de vídeo, por ejemplo, el codificador de vídeo 20, señala el factor de ponderación si la CU actual es una CU inter-codificada que no pertenece a una imagen IRAP. En tales ejemplos, cuando una imagen es una imagen de acceso aleatorio, el codificador de vídeo nunca transmite el factor de ponderación. En otros ejemplos, el codificador de vídeo 20 señala adicionalmente que el factor de ponderación es para las CU inter-codificadas pertenecientes a una imagen IRAP, si al menos una de sus imágenes de referencia (que solo puede ser una imagen de referencia de inter-vista) tiene una imagen de referencia

de inter-vista en cualquiera de sus listas de imágenes de referencia. En dichos ejemplos, los codificadores de vídeo pueden realizar ARP de la predicción residual de inter-vista para imágenes dentro de la unidad de acceso

- 5 Para ejemplos en los que un codificador de vídeo, por ejemplo, el codificador de vídeo 20, señala el factor de ponderación si la CU actual es una CU inter-codificada que no pertenece a una imagen IRAP, se cambia la tabla de sintaxis de la coding_unit, como se resalta a continuación. Las adiciones relativas a 3D-HEVC Test Model 4 están subrayadas y las supresiones se muestran con texto.

coding_unit(x0, y0, log2CbSize, ctDepth) {	Descriptor
if(transquant_bypass_enable_flag) {	
cu_transquant_bypass_flag	ae(v)
}	
if(slice_type != I)	
skip_flag[x0][y0]	ae(v)
if(skip_flag[x0][y0]) {	
prediction_unit (x0, y0, log2CbSize)	
si (iv_res_pred_flag && <u>(nal unit type > =</u> <u>BLA W LP && nal unit type < = RSV IRAP VCL23)</u>)	
iv_res_pred_weight_idx	ae(v)
si (icEnableFlag)	
ic_flag	ae(v)
}	
else {	
nCbS = (1 << log2CbSize)	
if(slice_type != I)	
pred_mode_flag	ae(v)
if((PredMode[x0][y0] != MODE_INTRA log2CbSize == Log2MinCbSize) && !predPartModeFlag)	
part_mode	ae(v)
if(PredMode[x0][y0] == MODE_INTRA) {	
if(PartMode == PART_2Nx2N && pcm_enabled_flag && log2CbSize >= Log2MinIPCMCUsSize && log2CbSize <= Log2MaxIPCMCUsSize)	
pcm_flag	ae(v)
if(pcm_flag) {	
num_subsequent_pcm	tu(3)
NumPCMBlock = num_subsequent_pcm + 1	
while(!byte_aligned())	
pcm_alignment_zero_bit	f(1)
pcm_sample(x0, y0, log2CbSize)	
} else {	
pbOffset = (PartMode == PART_NxN) ? (nCbS / 2) : 0	
log2PbSize = log2CbSize - (PartMode == PART_NxN ? 1 : 0)	
for(j = 0; j <= pbOffset; j = j + pbOffset)	
for(i = 0; i <= pbOffset; i = i + pbOffset) {	
if(vps_depth_modes_flag[nub_layer_id])	
depth_mode_parameters(x0 + i, y0 + j)	
if(depth_intra_mode[x0 + i][y0 + j] == INTRA_DEP_NONE)	
prev_intra_luma_pred_flag[x0 + i][y0 + j]	ae(v)
}	
for(j = 0; j <= pbOffset; j = j + pbOffset)	
for(i = 0; i <= pbOffset; i = i + pbOffset) {	
if(depth_intra_mode[x0 + i][y0 + j] == INTRA_DEP_NONE) {	

coding_unit(x0, y0, log2CbSize, ctDepth) {	Descriptor
if(prev_intra_luma_pred_flag[x0 + i][y0 + j])	
mpm_idx[x0 + i][y0 + j]	ae(v)
else	
rem_intra_luma_pred_mode[x0 + i][y0 + j]	ae(v)
}	
}	
if (!SdcFlag[x0][y0])	
intra_chroma_pred_mode[x0][y0]	ae(v)
}	
} else {	
if(PartMode == PART_2Nx2N) {	
if (iv_res_pred_flag && (<u>nal unit type > = BLA W LP && nal unit type < = RSV</u> <u>IRAP VCL23</u>))	
iv_res_pred_weight_idx	ae(v)
prediction_unit(x0, y0, nCbS, nCbS)	
} else if(PartMode == PART_2NxN) {	
...	
}	

Además, la variable TempRefPicInListsFlag y el proceso de obtención relacionado de TempRefPicInListsFlag se elimina como se muestra a continuación:

5 H.8.3.7 Proceso de obtención para el índice de referencia objetivo alternativo para TMVP en el modo de fusión

- Este proceso se invoca cuando el fragmento actual es un fragmento P o B.
- 10 • Las variables AltRefIdxL0 y AltRefIdxL1 se establecen igual a -1 y lo siguiente se aplica para X en el rango de 0 a 1, inclusive:
- Cuando X es igual a 0 o el fragmento actual es un fragmento B, se aplica lo siguiente:
- 15 ▪ ¿zeroldxLtFlag = RefPicListX[0] es una imagen de referencia a corto plazo? 0 : 1
- for (i = 1; i <= num_ref_idx_1X_active_minus 1 && AltRefIdxLX == -1; i++)
- if ((zeroldxLtFlag && RefPicListX[i] es una imagen de referencia a corto plazo) ||
- o (!zeroldxLtFlag && RefPicListX[i] es una imagen de referencia a largo plazo))
- 20 o AltRefIdxLX = i

Las propuestas existentes para ARP temporal deshabilitan ARP cuando NBDV no devuelve un DV disponible para el bloque de vídeo actual. Sin embargo, como se ha analizado anteriormente, esta divulgación proporciona técnicas para ARP que no dependen de un DV obtenido mediante NBDV. En consecuencia, en algunos ejemplos de acuerdo con esta divulgación, en lugar de inhabilitar siempre ARP cuando NBDV no devuelve un DV disponible, un

25 codificador de vídeo puede habilitar ARP en al menos algunas situaciones en las que NBDV no devuelve un DV disponible. Por ejemplo, un codificador de vídeo, p. ej., el codificador de vídeo 20 y/o el decodificador de vídeo 30, puede habilitar un ARP temporal si un bloque de vídeo de referencia temporal (**CurrTref**) cubre al menos un DMV. Como otro ejemplo, un codificador de vídeo puede habilitar un ARP temporal si el bloque de vídeo de referencia temporal (**CurrTref**) cubre al menos un DMV y el bloque correspondiente no está codificado con el modo BVSP. En dichos ejemplos, el codificador de vídeo puede aplicar un ARP temporal usando el DMV para reemplazar el DV, por ejemplo, como se ha descrito anteriormente con respecto a la FIG. 11. Como otro ejemplo, un codificador de vídeo puede habilitar un ARP de inter-vista, por ejemplo, como se ha descrito anteriormente con respecto a la FIG. 10, si la imagen de referencia actual es una imagen de referencia de inter-vista. Pueden darse una o más restricciones en el

30 decodificador de vídeo para que cuando NBDV no devuelva un DV disponible y una o más de las condiciones anteriores no sean verdaderas, el factor de ponderación w, para ARP se ponga a 0.

La FIG. 14 es un diagrama de bloques que ilustra un codificador de vídeo a modo de ejemplo 20 que puede implementar las técnicas descritas en esta divulgación. El codificador de vídeo 20 puede realizar una intra e inter-codificación de bloques de vídeo dentro de fragmentos de vídeo. La intra-codificación se apoya en la predicción

40

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 o inter-vista para reducir o eliminar la redundancia temporal en el vídeo dentro de tramas o imágenes adyacentes de una secuencia de vídeo. El modo intra (modo I) puede referirse a cualquiera de varios modos de compresión espacial. Las inter-modalidades, tales como la predicción unidireccional (modalidad P) o la bi-predicción (modalidad B), pueden incluir cualquiera de varias modalidades de compresión de base temporal.

En el ejemplo de la FIG. 14, el codificador de vídeo 20 incluye una unidad de división 235, una unidad de procesamiento de predicción 241, una memoria de imágenes de referencia 264, un sumador 250, una unidad de procesamiento de transformación 252, una unidad de procesamiento de cuantificación 254 y una unidad de codificación por entropía 256. La unidad de procesamiento de predicción 241 incluye una unidad de estimación de movimiento 242, una unidad de compensación de movimiento 244, una unidad de predicción residual avanzada (ARP) 254 y una unidad de procesamiento de intra-predicción 246. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 incluye además una unidad de procesamiento de cuantificación inversa 258, una unidad de procesamiento de transformación inversa 260 y un sumador 262. También puede incluirse un filtro de desbloqueo (no se muestra en la FIG. 14) para filtrar límites de bloque, para eliminar distorsiones de efecto pixelado del vídeo reconstruido. Si se desea, el filtro de desbloqueo filtrará típicamente la salida del sumador 262. También pueden usarse filtros de bucle adicionales (en bucle o tras el bucle), además del filtro de desbloqueo.

En varios ejemplos, una unidad de codificador de vídeo 20 puede estar encargada de realizar las técnicas de esta divulgación. Además, en algunos ejemplos, las técnicas de esta divulgación pueden dividirse entre una o más de las unidades del codificador de vídeo 20. Por ejemplo, la unidad ARP 245 puede realizar las técnicas de esta divulgación, solas o en combinación con otras unidades de codificador de vídeo, tales como la unidad de estimación de movimiento 242 y la unidad de compensación de movimiento 244.

Como se muestra en la FIG. 14, el codificador de vídeo 20 recibe datos de vídeo y la unidad de división 235 divide los datos en bloques de vídeo. Esta división también puede incluir la división en fragmentos, elementos u otras unidades mayores, así como la división de bloques de vídeo, por ejemplo, de acuerdo con una estructura de árbol cuaternario de unas LCU y CU. El codificador de vídeo 20 ilustra en general los componentes que codifican bloques de vídeo de un fragmento de vídeo que se va a codificar. El fragmento puede dividirse en múltiples bloques de vídeo (y, posiblemente, en conjuntos de bloques de vídeo denominados elementos).

La unidad de procesamiento de predicción 241 puede seleccionar una entre una pluralidad de posibles modos de codificación, tal como una entre una pluralidad de modos de intra-codificación, o una de una pluralidad de modos de inter-codificación, 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 241 puede proporcionar el bloque intra-codificado o inter-codificado resultante al sumador 250 para generar datos de bloques residuales, y al sumador 262 para reconstruir el bloque codificado para su uso como una imagen de referencia.

La unidad de intra-predicción 246, dentro de la unidad de procesamiento de predicción 241, puede realizar la codificación intra-predictiva del bloque de vídeo actual con respecto a uno o más bloques contiguos en la misma trama o fragmento que el bloque que va a codificarse, para proporcionar compresión espacial. La unidad de estimación de movimiento 242 y la unidad de compensación de movimiento 244, dentro de la unidad de procesamiento de predicción 241, realizan la codificación inter-predictiva del bloque de vídeo actual con respecto a uno o más bloques predictivos en una o más imágenes de referencia, por ejemplo para proporcionar compresión temporal.

La unidad de estimación de movimiento 242 puede estar configurada para determinar el modo de inter-predicción para un fragmento de vídeo de acuerdo con un patrón predeterminado para una secuencia de vídeo. La unidad de estimación de movimiento 242 y la unidad de compensación de movimiento 244 pueden estar sumamente integradas, pero se ilustran por separado con fines conceptuales. La estimación de movimiento, realizada por la unidad de estimación de movimiento 242, es el proceso de generación de vectores de movimiento, que estiman el movimiento de 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 con respecto a un bloque predictivo de una imagen de referencia.

Un bloque predictivo es un bloque del que se descubre que se corresponde estrechamente con la PU del bloque de vídeo que se va a codificar en términos de diferencia de píxeles, que puede determinarse mediante la suma de una diferencia absoluta (SAD), suma de diferencia al cuadrado (SSD) u otras métricas de diferencia. En algunos ejemplos, el codificador de vídeo 20 puede calcular valores para posiciones de píxel de subentero de imágenes de referencia almacenadas en la memoria de imágenes de referencia 264. Por ejemplo, el codificador de vídeo 20 puede interpolar valores de posiciones de un cuarto de píxel, posiciones de un octavo de píxel u otras posiciones de píxel fraccionarias de la imagen de referencia. Por lo tanto, la unidad de estimación de movimiento 242 puede realizar una búsqueda de movimiento con respecto a las posiciones de píxel completo y a las posiciones de píxel fraccionario, y emitir un vector de movimiento con una precisión de píxel fraccionario.

La unidad de estimación de movimiento 242 calcula un vector de movimiento para una PU de un bloque de vídeo de un fragmento sometido a intercodificación, comparando la posición de la PU con la posición de un bloque predictivo de una imagen de referencia. La imagen de referencia puede seleccionarse entre una primera lista de imágenes de referencia (lista 0 o RefPicList0) o una segunda lista de imágenes de referencia (lista 1 o RefPicList1), cada una de las cuales identifica una o más imágenes de referencia almacenadas en la memoria de imágenes de referencia 264. La unidad de estimación de movimiento 242 envía el vector de movimiento calculado a la unidad de codificación por entropía 256 y a la unidad de compensación de movimiento 246.

La compensación de movimiento, realizada por la unidad de compensación de movimiento 244, puede implicar extraer o generar el bloque predictivo basándose en el vector de movimiento determinado por la estimación de movimiento, realizando posiblemente interpolaciones hasta la precisión de subpíxel. Tras recibir el vector de movimiento para la PU del bloque de vídeo actual, la unidad de compensación de movimiento 244 puede localizar el bloque predictivo al que apunta el vector de movimiento en una de las listas de imágenes de referencia. El codificador de vídeo 20 forma un bloque de vídeo residual restando los valores de píxeles del bloque predictivo a los valores de píxeles del bloque de vídeo actual que se está codificando, generando valores de diferencia de píxel. Los valores de diferencia de píxel forman datos residuales para el bloque, y pueden incluir componentes de diferencia de luma y croma. El sumador 250 representa el componente o los componentes que realizan esta operación de sustracción. La unidad de compensación de movimiento 244 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.

La unidad de intra-predicción 246 puede intra-predecir un bloque actual, como alternativa a la inter-predicción llevada a cabo por la unidad de estimación de movimiento 242 y la unidad de compensación de movimiento 244, como se ha descrito anteriormente. En particular, la unidad de procesamiento de intra-predicción 246 puede determinar un modo de intra-predicción a usar para codificar un bloque actual. En algunos ejemplos, la unidad de procesamiento de intra-predicción 246 puede codificar un bloque de vídeo actual usando varias modalidades de intra-predicción, por ejemplo, durante diferentes pasadas de codificación, y el módulo de intra-predicción 246 (o la unidad de procesamiento de predicción 241, en algunos ejemplos) puede seleccionar una modalidad adecuada de intra-predicción a usar, entre las modalidades probadas. Por ejemplo, la unidad de procesamiento de intra-predicción 246 puede calcular valores de velocidad-distorsión usando un análisis de velocidad-distorsión para los diversos modos de intra-predicción probadas, y seleccionar el modo de intra-predicción que tenga las mejores características de velocidad-distorsión entre los modos probados. El análisis de velocidad-distorsión determina en general una cantidad de distorsión (o error) entre un bloque codificado y un bloque original no codificado que se codificó para generar el bloque codificado, así como una tasa de bits (es decir, un número de bits) usada para generar el bloque codificado. La unidad de procesamiento de intra-predicción 246 puede calcular relaciones a partir de las distorsiones y velocidades para los diversos bloques codificados, para determinar qué modo de intra-predicción presenta el mejor valor de velocidad-distorsión para el bloque.

En cualquier caso, tras seleccionar un modo de intra-predicción para un bloque, la unidad de procesamiento de intra-predicción 246 puede proporcionar información que indica el modo de intra-predicción seleccionada para el bloque, a la unidad de codificación por entropía 256. La unidad de codificación por entropía 256 puede codificar la información que indica la modalidad de intra-predicción seleccionada de acuerdo con las técnicas de esta divulgación. El codificador de vídeo 20 puede incluir datos de configuración en el flujo de bits transmitidos, que pueden incluir una pluralidad de tablas de índices de modalidades de intra-predicción y una pluralidad de tablas de índices de modalidades de intra-predicción modificadas (también denominadas tablas de asignación de palabras de código), definiciones de contextos de codificación para varios bloques e indicaciones de la modalidad de intra-predicción más probable, una tabla de índices de modalidades de intra-predicción y una tabla modificada de índices de modalidades de intra-predicción, a usar en cada uno de los contextos.

Después de que la unidad de procesamiento de predicción 241 genera el bloque predictivo para el bloque de vídeo actual, ya sea mediante la inter-predicción o la intra-predicción, 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 del bloque residual pueden incluirse en una o más TU y aplicarse a la unidad de procesamiento de transformación 252. La unidad de procesamiento de transformación 252 transforma los datos de vídeo residuales en coeficientes de transformación residuales mediante una transformación, tal como una transformación de coseno discreta (DCT) o una transformación similar desde un punto de vista conceptual. La unidad de procesamiento de transformación 252 puede convertir los datos de vídeo residuales de un dominio de píxel a un dominio de transformación, tal como un dominio de frecuencia.

La unidad de procesamiento de transformación 252 puede enviar los coeficientes de transformación resultantes a la unidad de procesamiento de cuantificación 254. La unidad de procesamiento de cuantificación 254 cuantifica los coeficientes de transformación para reducir adicionalmente la velocidad binaria. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos o la totalidad de los coeficientes. El grado de cuantificación puede modificarse ajustando un parámetro de cuantificación. En algunos ejemplos, la unidad de procesamiento de cuantificación 254 puede llevar a cabo un escaneado de la matriz, incluyendo los coeficientes de transformación cuantificados. De forma alternativa, la unidad de codificación por entropía 256 puede llevar a cabo el escaneado.

Tras la cuantificación, la unidad de codificación por entropía 256 realiza la codificación por entropía de los coeficientes de transformación cuantificados. Por ejemplo, la unidad de codificación por entropía 256 puede realizar una codificación de longitud variable adaptativa según el contexto (CAVLC), una codificación aritmética binaria adaptativa según el contexto (CABAC), una codificación aritmética binaria adaptativa según el contexto basándose 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 realizada por la unidad de codificación por entropía 256, el flujo de bits de vídeo codificado puede transmitirse al descodificador de vídeo 30, o archivarse para su posterior transmisión o recuperación mediante el descodificador de vídeo 30. La unidad de codificación por entropía 256 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.

La unidad de procesamiento de cuantificación inversa 258 y la unidad de procesamiento de transformación inversa 260 aplican una cuantificación inversa y una transformación inversa, respectivamente, para reconstruir el bloque residual en el dominio de píxel, para su posterior uso como un bloque de referencia de una imagen de referencia. La unidad de compensación de movimiento 244 puede calcular un bloque de referencia añadiendo el bloque residual a un bloque predictivo de una de las imágenes de referencia de una de las listas de imágenes de referencia. La unidad de compensación de movimiento 244 también puede aplicar uno o más filtros de interpolación al bloque residual reconstruido para calcular valores de píxel subentero y usarlos en la estimación de movimiento. El sumador 262 añade el bloque residual reconstruido al bloque predictivo con compensación de movimiento generado por la unidad de compensación de movimiento 244 para generar un bloque de referencia para su almacenamiento en la memoria de imágenes de referencia 264. La unidad de estimación de movimiento 242 y la unidad de compensación de movimiento 244 pueden usar el bloque de referencia como un bloque de referencia para realizar la inter-predicción de un bloque en una trama o imagen de vídeo posterior.

El codificador de vídeo 20, por ejemplo, la unidad ARP 245 del codificador de vídeo 20, puede realizar cualquiera de las técnicas de ARP, por ejemplo las técnicas de ARP temporales o de inter-venta descritas aquí. Por ejemplo, si la unidad de procesamiento de predicción 241 y/o la unidad de estimación de movimiento 242 inter-venta predice un bloque de vídeo actual, por ejemplo, predice el bloque de vídeo actual basándose en un bloque de referencia de una imagen de referencia en una vista de referencia diferente al bloque de vídeo actual utilizando un DMV, la unidad ARP 245 puede identificar el DMV asociado con el bloque de vídeo actual para la predicción de inter-venta del bloque de vídeo actual. En algunos ejemplos, el DMV puede ser un DV convertido a un IDMV para la predicción de información de movimiento del bloque de vídeo actual.

Basándose en el DMV, la unidad ARP 245, sola o con la unidad de compensación de movimiento 244, también puede identificar el bloque de vídeo de referencia de inter-venta (**Base**) y un TMV del bloque de vídeo de referencia de inter-venta, que puede haberse determinado previamente mediante la unidad de estimación de movimiento 242 durante la predicción del bloque de vídeo de referencia de inter-venta (**Base**). Basándose en el TMV, la unidad ARP 245, sola o con unidad de compensación de movimiento 244, puede identificar un bloque de vídeo de referencia temporal en la vista de referencia (**BaseTRef**) y un bloque de vídeo de referencia temporal en la vista actual (**CurrTRef**). La unidad ARP 245 puede determinar el predictor residual de inter-venta para el bloque de vídeo actual basándose en la diferencia (**CurrTRef - BaseTRef**) entre el bloque de vídeo de referencia temporal en la vista actual y el bloque de vídeo de referencia temporal en la vista de referencia. La unidad ARP 245 puede aplicar un factor de ponderación w a la diferencia (**CurrTRef - BaseTRef**), y puede determinar que el bloque de predicción de inter-venta para el bloque de vídeo actual sea **Base + w (CurrTRef - BaseTRef)**, como se describe en el presente documento, por ejemplo, con respecto a la FIG. 10.

Como otro ejemplo, si la unidad de procesamiento de predicción 241 y/o la unidad de estimación de movimiento 242 predice temporalmente un bloque de vídeo actual, por ejemplo, predice el bloque de vídeo actual basándose en un bloque de referencia de una imagen de referencia en una unidad de acceso diferente a, pero con la misma vista que el bloque de vídeo actual usando un TMV, la unidad ARP 245 puede identificar el TMV. Basándose en el TMV, la unidad ARP 245, sola o con unidad de compensación de movimiento 244, también puede identificar el bloque de vídeo de referencia temporal (**CurrTRef**) y un DMV del bloque de vídeo de referencia temporal, que puede haberse determinado previamente mediante la unidad de estimación de movimiento 242 durante la predicción del bloque de vídeo de referencia temporal (**CurrTRef**). Basándose en el DMV, la unidad ARP 245, sola o con unidad de compensación de movimiento 244, puede identificar un bloque de vídeo de referencia temporal en la vista de referencia (**BaseTRef**) y un bloque de vídeo de referencia de inter-venta en la vista de referencia (**Base**). La unidad ARP 245 puede determinar el predictor residual temporal para el bloque de vídeo actual basándose en la diferencia (**Base - BaseTRef**) entre los bloques de vídeo de referencia en la vista de referencia. La unidad ARP 245 puede aplicar un factor de ponderación w a la diferencia (**Base - BaseTRef**), y puede determinar que el bloque de predicción temporal para que el bloque de vídeo actual sea **CurrTRef + w * (Base - BaseTRef)**, como se describe en el presente documento, por ejemplo, con respecto a la FIG. 11.

En cualquiera de los ejemplos anteriores, la unidad ARP 245, la unidad de compensación de movimiento 244 y/o cualquier componente de la unidad de procesamiento de predicción 241 o el codificador de vídeo 20, pueden proporcionar el bloque de predicción de inter-venta al sumador 250, que determina el valor residual que se va a

codificar en el flujo de bits de vídeo codificado para el bloque de vídeo actual. Adicionalmente, la unidad ARP 245 puede escalar TMVs y DMVs, o llevar a cabo cualquiera de las funciones descritas en el presente documento para ARP de acuerdo con las técnicas de esta divulgación.

- 5 De esta manera, el codificador de vídeo 20 puede estar configurado para implementar las técnicas ARP de ejemplo de esta divulgación para codificar un bloque de vídeo. Por ejemplo, el codificador de vídeo 20 puede ser un ejemplo de un codificador de vídeo configurado para realizar un procedimiento de predicción residual avanzada de inter-vista para codificar datos de vídeo que comprende identificar un DMV para un bloque de vídeo de vídeo actual, en el que el bloque de vídeo actual está en una vista actual, y en el que el DMV se utiliza para la predicción de inter-vista del bloque de vídeo actual basándose en un bloque de vídeo de referencia de inter-vista en una vista de referencia y en la misma unidad de acceso que el bloque de vídeo actual. El procedimiento comprende además identificar un vector de movimiento temporal (TMV) y una imagen de referencia asociada del bloque de vídeo de referencia de inter-vista, identificando un bloque de vídeo de referencia temporal en la imagen de referencia asociada en la vista de referencia basándose en el TMV del bloque de vídeo de referencia de inter-vista, e identificar un bloque de vídeo de referencia temporal en la vista actual basándose en el TMV del bloque de vídeo de referencia de inter-vista en la vista de referencia. El bloque de vídeo de referencia temporal en la vista actual y el bloque de vídeo de referencia temporal en la vista de referencia están en la misma unidad de acceso. El procedimiento comprende además determinar un bloque de predicción residual para el bloque de vídeo actual basándose en una diferencia entre el bloque de vídeo de referencia temporal en la vista actual y el bloque de vídeo de referencia temporal en la vista de referencia, y codificar un flujo de bits de vídeo codificado que codifica los datos de vídeo para identificar el DMV y un bloque residual para el bloque de vídeo actual, en el que el bloque residual identificado por el flujo de bits de vídeo codificado comprende una diferencia entre el bloque de vídeo de referencia de inter-vista y el bloque de predicción residual para el bloque de vídeo actual.
- 10
- 15
- 20
- 25 El codificador de vídeo 20 también puede ser un ejemplo de un codificador de vídeo que comprende una memoria configurada para almacenar un flujo de bits de vídeo codificado que codifica los datos de vídeo y uno o más procesadores. El uno o más procesadores del codificador de vídeo, por ejemplo, el codificador de vídeo 20, pueden configurarse para identificar un DMV para un bloque de vídeo actual, en el que el bloque de vídeo actual está en una vista actual y en el que el DMV se utiliza para la predicción de inter-vista del bloque de vídeo actual basándose en un bloque de vídeo de referencia de inter-vista en una vista de referencia y en la misma unidad de acceso que el bloque de vídeo actual. El uno o más procesadores están configurados adicionalmente para identificar un vector de movimiento temporal (TMV) y una imagen de referencia asociada del bloque de vídeo de referencia de inter-vista, identificar un bloque de vídeo de referencia temporal en la imagen de referencia asociada en la vista de referencia basándose en el TMV del bloque de vídeo de referencia de inter-vista e identificar un bloque de vídeo de referencia temporal en la vista actual basándose en el TMV del bloque de vídeo de referencia de inter-vista en la vista de referencia. El bloque de vídeo de referencia temporal en la vista actual y el bloque de vídeo de referencia temporal en la vista de referencia están en la misma unidad de acceso. El uno o más procesadores están configurados adicionalmente para determinar un bloque de predicción residual para el bloque de vídeo actual basándose en una diferencia entre el bloque de vídeo de referencia temporal en la vista actual y el bloque de vídeo de referencia temporal en la vista de referencia, y codificar el flujo de bits de vídeo codificado para identificar el DMV y un bloque residual para el bloque de vídeo actual, en el que el bloque residual identificado codificando el flujo de bits de vídeo codificado comprende una diferencia entre el bloque de vídeo de referencia de inter-vista y el bloque de predicción residual para el bloque de vídeo actual.
- 30
- 35
- 40
- 45 La FIG. 15 es un diagrama de bloques que ilustra un descodificador de vídeo a modo de ejemplo 30 que puede implementar las técnicas descritas en esta divulgación. En el ejemplo de la FIG. 15, el descodificador de vídeo 30 incluye una unidad de descodificación por entropía 280, una unidad de procesamiento de predicción 281, una unidad de procesamiento de cuantificación inversa 286, una unidad de transformación inversa 288, un sumador 290 y una memoria de imágenes de referencia 292. La unidad de procesamiento de predicción 281 incluye la unidad de compensación de movimiento 282, la unidad ARP 283 y la unidad de intra-predicción 284. En algunos ejemplos, el descodificador de vídeo 30 puede realizar una pasada de descodificación en general recíproca a la pasada de codificación descrita con respecto al codificador de vídeo 20 de la FIG. 14.
- 50
- 55 En varios ejemplos, una unidad de descodificador de vídeo 30 puede ser responsable de la realización de las técnicas de esta divulgación. Además, en algunos ejemplos, las técnicas de esta divulgación pueden dividirse entre una o más de las unidades del descodificador de vídeo 30. Por ejemplo, la unidad ARP 283 puede realizar las técnicas de esta divulgación, solas o en combinación con otras unidades de codificador de vídeo, tal como la unidad de compensación de movimiento 282.
- 60 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 280 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 280 remite los vectores de movimiento y otros elementos sintácticos a la unidad de procesamiento de predicción 281. El descodificador de vídeo 30 puede recibir los elementos sintácticos al nivel del fragmento de vídeo y/o el nivel del bloque de vídeo.
- 65

Quando el fragmento de vídeo se codifica como un fragmento intra-codificado (I), la unidad de intra-predicción 284 de la unidad de procesamiento de predicción 281 puede generar datos de predicción para un bloque de vídeo del fragmento de vídeo actual, basándose en un modo de intra-predicción indicado y unos datos de bloques
 5 previamente descodificados de la trama o imagen actual. Cuando la trama de vídeo es codificada como un fragmento inter-codificado (es decir, B o P), la unidad de compensación de movimiento 282 de la unidad de procesamiento de predicción 281 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 280. Los bloques predictivos pueden ser generados a partir de una de las imágenes de referencia
 10 dentro de una de las listas de imágenes de referencia. El descodificador de vídeo 30 puede construir las listas de tramas de referencia, RefPicList0 and RefPicList1 usando técnicas de construcción por defecto o cualquier otra técnica basándose en las imágenes de referencia almacenadas en la memoria de imágenes de referencia 292.

La unidad de compensación de movimiento 282 determina la información de predicción para un bloque de vídeo del
 15 fragmento de vídeo actual, analizando los vectores de movimiento y otros elementos sintácticos, y usa la información de predicción para generar los bloques predictivos del bloque de vídeo actual que está siendo descodificado. Por ejemplo, la unidad de compensación de movimiento 282 usa algunos de los elementos sintácticos recibidos para determinar una modalidad de predicción (por ejemplo, intra-predicción o inter-predicción), usada para codificar los bloques de vídeo del fragmento de vídeo, un tipo de fragmento de inter-predicción (por ejemplo, fragmento B o
 20 fragmento P), información de construcción para una o más de las listas de imágenes de referencia del fragmento, vectores de movimiento para cada bloque de vídeo inter-codificado del fragmento, el estado de inter-predicción para cada bloque de vídeo inter-codificado del fragmento y otra información, para descodificar los bloques de vídeo en el fragmento de vídeo actual.

La unidad de compensación de movimiento 282 también puede realizar la interpolación basándose en filtros de
 25 interpolación. La unidad de compensación de movimiento 282 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 de píxeles subenteros de los bloques de referencia. En este caso, la unidad de compensación de movimiento 282 puede determinar los filtros de interpolación usados por el codificador de vídeo 20 a partir de los elementos
 30 sintácticos recibidos y usar los filtros de interpolación para generar bloques predictivos.

La unidad de procesamiento de cuantificación inversa 286 cuantifica de manera inversa, es decir, descuantifica, los
 coeficientes de transformación cuantificados, proporcionados en el flujo de bits y descodificados por la unidad de
 35 descodificación por entropía 280. 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 en el fragmento de vídeo, para determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa que debería aplicarse. La unidad de procesamiento de transformación inversa 288 aplica una transformación inversa, por ejemplo una DCT
 inversa, una transformación inversa entera o un proceso de transformación inversa conceptualmente similar, a los
 40 coeficientes de transformación, con el fin de generar bloques residuales en el dominio de píxeles.

Después de que la unidad de compensación de movimiento 282 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
 45 transformación inversa 288 con los correspondientes bloques predictivos generados por la unidad de compensación de movimiento 282. El sumador 290 representa el componente o los componentes que llevan a cabo esta operación de suma. Si se desea, también puede aplicarse un filtro de desbloqueo para filtrar los bloques descodificados con el fin de eliminar distorsiones de efecto pixelado. También pueden utilizarse otros 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
 50 calidad del vídeo. Los bloques de vídeo descodificados de una trama o imagen dada son a continuación almacenados en la memoria de imágenes de referencia 292, que almacena imágenes de referencia usadas para la posterior compensación de movimiento. La memoria de imágenes de referencia 292 almacena también vídeo descodificado para su presentación posterior en un dispositivo de visualización, tal como el dispositivo de
 visualización 32 de la FIG. 1.

El descodificador de vídeo 30, por ejemplo, la unidad ARP 283 del descodificador de vídeo 30, puede realizar
 55 cualquiera de las técnicas ARP, por ejemplo, las técnicas de ARP temporales o de inter-vista descritas en el presente documento. Por ejemplo, si, basándose en elementos sintácticos recuperados del flujo de bits de vídeo codificado por la unidad de descodificación por entropía 280, la unidad de procesamiento de predicción 281 y/o la unidad de compensación de movimiento 282 realiza una predicción de inter-vista de un bloque de vídeo actual
 60 usando un DMV, la unidad ARP 283 puede identificar el DMV asociado con el bloque de vídeo actual para la predicción de inter-vista del bloque de vídeo actual. En algunos ejemplos, el DMV puede ser un DV convertido a un IDMV para la predicción de información de movimiento del bloque de vídeo actual.

Basándose en el DMV, la unidad ARP 283, sola o con la unidad de compensación de movimiento 282, también
 65 puede identificar el bloque de vídeo de referencia de inter-vista (**Base**) y un TMV del bloque de vídeo de referencia de inter-vista, que puede haberse previamente determinado mediante la unidad de compensación de movimiento

282 durante la predicción del bloque de vídeo de referencia de inter-vista (**Base**). Basándose en el TMV, la unidad ARP 283, sola o con unidad de compensación de movimiento 282, puede identificar un bloque de vídeo de referencia temporal en la vista de referencia (**BaseTRef**) y un bloque de vídeo de referencia temporal en la vista actual (**CurrTRef**). La unidad ARP 283 puede determinar el predictor residual de inter-vista para el bloque de vídeo actual basándose en la diferencia (**CurrTRef - BaseTRef**) entre el bloque de vídeo de referencia temporal en la vista actual y el bloque de vídeo de referencia temporal en la vista de referencia. La unidad ARP 283 puede aplicar un factor de ponderación w a la diferencia (**CurrTRef - BaseTRef**), y puede determinar que el bloque de predicción de inter-vista para que el bloque de vídeo actual sea **Base + w*(CurrTRef - BaseTRef)**, como se describe en el presente documento, por ejemplo con respecto a la FIG. 10.

Como otro ejemplo, si, basándose en elementos sintácticos recuperados del flujo de bits de vídeo codificado por la unidad de descodificación por entropía 280, la unidad de procesamiento de predicción 281 y/o la unidad de compensación de movimiento 282 predice temporalmente un bloque de vídeo actual usando un TMV, la unidad ARP 283 puede identificar el TMV. Basándose en el TMV, la unidad ARP 283, sola o con la unidad de compensación de movimiento 282, también puede identificar el bloque de vídeo de referencia temporal (**CurrTRef**) y un DMV del bloque de vídeo de referencia temporal, que puede haberse previamente determinado mediante la unidad de compensación de movimiento 282 durante la predicción del bloque de vídeo de referencia temporal (**CurrTRef**). Basándose en el DMV, la unidad ARP 283, sola o con la unidad de compensación de movimiento 282, puede identificar un bloque de vídeo de referencia temporal en la vista de referencia (**BaseTRef**) y un bloque de vídeo de referencia de inter-vista en la vista de referencia (**Base**). La unidad ARP 283 puede determinar el predictor residual temporal para el bloque de vídeo actual basándose en la diferencia (**Base - BaseTRef**) entre los bloques de vídeo de referencia en la vista de referencia. La unidad ARP 283 puede aplicar un factor de ponderación w a la diferencia (**Base - BaseTRef**), y puede determinar que el bloque de predicción temporal para el bloque de vídeo actual sea **CurrTRef + w*(Base - BaseTRef)**, como se describe en el presente documento, por ejemplo, con respecto a la FIG. 11.

En cualquiera de los ejemplos anteriores, la unidad ARP 283, la unidad de compensación de movimiento 282 y/o cualquier componente de la unidad de procesamiento de predicción 281 o el descodificador de vídeo 30, pueden proporcionar el bloque de predicción de inter-vista al sumador 290, que suma el bloque de predicción con el valor residual descodificado recibido de la unidad de procesamiento de transformación inversa 288 para reconstruir el bloque de vídeo actual. Adicionalmente, la unidad ARP 283 puede escalar TMVs y DMVs, o llevar a cabo cualquiera de las funciones descritas en el presente documento para ARP de acuerdo con las técnicas de esta divulgación.

De esta manera, el descodificador de vídeo 30 puede configurarse para implementar las técnicas ARP de ejemplo de esta divulgación para descodificar un bloque de vídeo. Por ejemplo, el descodificador de vídeo 30 puede ser un ejemplo de un descodificador de vídeo configurado para realizar un procedimiento de predicción residual avanzada de inter-vista para descodificar datos de vídeo que comprende descodificar un flujo de bits de vídeo codificado que codifica los datos de vídeo para identificar un vector de movimiento de disparidad (DMV) y un bloque residual para un bloque de vídeo actual, en el que el bloque de vídeo actual está en una vista actual y en el que el DMV se utiliza para la predicción de inter-vista del bloque de vídeo actual basándose en un bloque de vídeo de referencia de inter-vista en una vista de referencia y en una misma unidad de acceso que el bloque de vídeo actual. El procedimiento comprende además identificar un vector de movimiento temporal (TMV) y una imagen de referencia asociada del bloque de vídeo de referencia de inter-vista, identificando un bloque de vídeo de referencia temporal en la imagen de referencia asociada en la vista de referencia basándose en el TMV del bloque de vídeo de referencia de inter-vista, e identificar un bloque de vídeo de referencia temporal en la vista actual basándose en el TMV del bloque de vídeo de referencia de inter-vista en la vista de referencia. El bloque de vídeo de referencia temporal en la vista actual y el bloque de vídeo de referencia temporal en la vista de referencia están en la misma unidad de acceso. El procedimiento comprende además determinar un bloque de predicción residual para el bloque de vídeo actual basándose en una diferencia entre el bloque de vídeo de referencia temporal en la vista actual y el bloque de vídeo de referencia temporal en la vista de referencia, y aplicar el bloque de predicción residual y el bloque residual identificados desde el flujo de bits de vídeo codificado al bloque de vídeo de referencia de inter-vista para reconstruir el bloque de vídeo actual.

El descodificador de vídeo 30 puede ser también un ejemplo de un codificador de vídeo que comprende una memoria configurada para almacenar un flujo de bits de vídeo codificado que codifica los datos de vídeo y uno o más procesadores. El uno o más procesadores del codificador de vídeo, por ejemplo, el descodificador de vídeo 30, pueden configurarse para identificar un DMV para un bloque de vídeo actual, en el que el bloque de vídeo actual está en una vista actual, y en el que el DMV se utiliza para la predicción de inter-vista del bloque de vídeo actual basándose en un bloque de vídeo de referencia de inter-vista en una vista de referencia y en una misma unidad de acceso que el bloque de vídeo actual. El uno o más procesadores están configurados adicionalmente para identificar un vector de movimiento temporal (TMV) y una imagen de referencia asociada del bloque de vídeo de referencia de inter-vista, identificar un bloque de vídeo de referencia temporal en la imagen de referencia asociada en la vista de referencia basándose en el TMV del bloque de vídeo de referencia de inter-vista e identificar un bloque de vídeo de referencia temporal en la vista actual basándose en el TMV del bloque de vídeo de referencia de inter-vista en la vista de referencia. El bloque de vídeo de referencia temporal en la vista actual y el bloque de vídeo de referencia temporal en la vista de referencia están en la misma unidad de acceso. El uno o más procesadores están

configurados además para determinar un bloque de predicción residual para el bloque de vídeo actual basándose en una diferencia entre el bloque de vídeo de referencia temporal en la vista actual y el bloque de vídeo de referencia temporal en la vista de referencia, y codificar el flujo de bits de vídeo codificado para identificar el DMV y un bloque residual para el bloque de vídeo actual, en el que el bloque residual identificado codificando el flujo de bits de vídeo codificado comprende una diferencia entre el bloque de vídeo de referencia de inter-vista y el bloque de predicción residual para el bloque de vídeo actual.

La FIG. 16 es un diagrama de flujo que ilustra un ejemplo de procedimiento de ARP para descodificar un bloque de vídeo de acuerdo con las técnicas descritas en esta divulgación. El procedimiento a modo de ejemplo de la FIG. 16 puede ser realizado por un descodificador de vídeo, tal como el descodificador de vídeo 30, que puede incluir una unidad ARP 283.

De acuerdo con el procedimiento a modo de ejemplo de la FIG. 16, el descodificador de vídeo 30 descodifica un flujo de bits de vídeo codificado para identificar un bloque de vídeo de referencia y un bloque residual para el bloque de vídeo actual (300). Por ejemplo, la unidad de compensación de movimiento 282 puede identificar el bloque de vídeo de referencia basándose en un vector de movimiento indicado por la sintaxis descodificada por la unidad de descodificación por entropía 280 y la unidad de procesamiento de transformación inversa 288 puede proporcionar el bloque residual descodificado al sumador 290. El descodificador de vídeo 30, por ejemplo, la unidad ARP 283, identifica un DMV desde una vista actual del bloque de vídeo actual a una vista de referencia (302).

El descodificador de vídeo 30, por ejemplo, la unidad ARP 283, puede entonces determinar un bloque de predicción residual para descodificar el bloque de vídeo actual basándose en el DMV (304). Por ejemplo, si el bloque de vídeo actual tiene predicción de inter-vista, el descodificador de vídeo 30 puede determinar un bloque de predicción residual de inter-vista basándose en el DMV del bloque de vídeo actual utilizando técnicas de ARP de inter-vista, por ejemplo, como se describe con respecto a la FIG. 10. Si se predice temporalmente el bloque de vídeo actual, el descodificador de vídeo 30 puede determinar un bloque de predicción residual temporal basándose en el DMV de un bloque de vídeo de referencia temporal utilizando técnicas de ARP temporal, por ejemplo, como se describe con respecto a la FIG. 11. El descodificador de vídeo 30, por ejemplo, la unidad ARP 283 y/o el sumador 290, puede aplicar el bloque de predicción residual y el bloque residual descodificado al bloque de vídeo de referencia para reconstruir el bloque de vídeo actual (306).

La FIG. 17 es un diagrama de flujo que ilustra un ejemplo de un procedimiento de ARP de inter-vista para descodificar un bloque de vídeo predicho de inter-vista de acuerdo con las técnicas descritas en esta divulgación. El procedimiento a modo de ejemplo de la FIG. 17 puede ser realizado por un descodificador de vídeo, tal como el descodificador de vídeo 30, que puede incluir una unidad ARP 283.

De acuerdo con el procedimiento a modo de ejemplo de la FIG. 17, el descodificador de vídeo 30 descodifica un flujo de bits de vídeo codificado para identificar un DMV utilizado para la predicción de inter-vista del bloque de vídeo actual y un bloque residual (310). El descodificador de vídeo 30, por ejemplo, la unidad ARP 283, identifica el bloque de vídeo de referencia de inter-vista (**Base**) basándose en el DMV (312). El descodificador de vídeo 30, por ejemplo, la unidad ARP 283, también identifica un TMV y una imagen de referencia asociada del bloque de vídeo de referencia de inter-vista (**Base**) (314).

El descodificador de vídeo 30, por ejemplo, la unidad ARP 283, puede entonces identificar bloques de vídeo de referencia temporales en las vistas actuales y de referencia (**CurrTRef** y **BaseTRef**, respectivamente) basándose en el TMV, por ejemplo, utilizando las técnicas descritas anteriormente con respecto a la FIG. 10 (316). El descodificador de vídeo 30, por ejemplo, la unidad ARP 283, puede entonces determinar un bloque de predicción residual de inter-vista para el bloque de vídeo actual basándose en la diferencia entre estos bloques de vídeo de referencia temporales (**CurrTRef - BaseTRef**) (318). El descodificador de vídeo, por ejemplo, la unidad ARP 283 y/o el sumador 290, puede aplicar el bloque de predicción residual de inter-vista y el bloque residual descodificado al bloque de vídeo de referencia de inter-vista (**Base**) para reconstruir el bloque de vídeo actual (**Curr**) (320).

La FIG. 18 es un diagrama de flujo que ilustra un ejemplo de procedimiento de ARP temporal para descodificar un bloque de vídeo predicho temporalmente de acuerdo con las técnicas descritas en esta divulgación. El procedimiento a modo de ejemplo de la FIG. 18 puede ser realizado por un descodificador de vídeo, tal como el descodificador de vídeo 30, que puede incluir una unidad ARP 283.

De acuerdo con el procedimiento a modo de ejemplo de la FIG. 18, el descodificador de vídeo 30 descodifica el flujo de bits de vídeo codificado para identificar un bloque de vídeo de referencia temporal (**CurrTRef**) en la vista actual y un bloque residual para reconstruir el bloque de vídeo actual (330). El descodificador de vídeo 30, por ejemplo, la unidad de compensación de movimiento 282, puede identificar el bloque de vídeo de referencia temporal (**CurrTRef**) en la vista actual usando un TMV asociado con el bloque de vídeo actual como se determina a partir del flujo de bits de vídeo descodificado. El descodificador de vídeo 30, por ejemplo, la unidad ARP 283, puede identificar un DMV del bloque de vídeo de referencia temporal (**CurrTRef**), que a su vez puede identificar un bloque de vídeo de referencia temporal en la vista de referencia (**BaseTRef**) (332).

El descodificador de vídeo 30, por ejemplo, la unidad ARP 283, también puede identificar un bloque de vídeo de referencia de inter-vista en la vista de referencia (**Base**) basándose en el DMV del bloque de vídeo de referencia temporal (**CurrTRef**) en la vista actual (334). El descodificador de vídeo 30, por ejemplo, la unidad ARP 283, puede entonces determinar el bloque de predicción residual temporal para el bloque de vídeo actual basándose en la

5 diferencia entre estos bloques de vídeo de referencia en la vista de referencia (**Base - BaseTRef**) (336). El descodificador de vídeo, por ejemplo, la unidad ARP 283 y/o el sumador 290, puede aplicar el bloque de predicción residual temporal y el bloque residual descodificado al bloque de vídeo de referencia temporal (**CurrTRef**) para reconstruir el bloque de vídeo actual (**Curr**) (338).

10 La FIG. 19 es un diagrama de flujo que ilustra un ejemplo de procedimiento de ARP para codificar un bloque de vídeo de acuerdo con las técnicas descritas en esta divulgación. El procedimiento a modo de ejemplo de la FIG. 19 puede realizarse mediante un codificador de vídeo, tal como un codificador de vídeo 20, que puede incluir una unidad ARP 245.

15 De acuerdo con el procedimiento a modo de ejemplo de la FIG. 19, el codificador de vídeo 20, por ejemplo la unidad ARP 245, identifica un DMV desde una vista actual de un bloque de vídeo actual a una vista de referencia (340). El codificador de vídeo 20, por ejemplo, la unidad ARP 245, puede determinar entonces un bloque de predicción residual para codificar el bloque de vídeo actual basándose en el DMV (342). Por ejemplo, si el bloque de vídeo actual tiene predicción de inter-vista, el codificador de vídeo 20 puede determinar un bloque de predicción residual

20 de inter-vista basándose en el DMV del bloque de vista actual usando técnicas de ARP de inter-vista, por ejemplo, como se describe con respecto a la FIG. 10. Si se predice temporalmente el bloque de vídeo actual, el codificador de vídeo 20 puede determinar un bloque de predicción residual temporal basándose en el DMV de un bloque de vídeo de referencia temporal en la vista actual usando técnicas de ARP temporal, por ejemplo, como se describe con respecto a la FIG. 11. En cualquier caso, el codificador de vídeo 20, por ejemplo, la unidad ARP 245 y el sumador

25 250 pueden determinar un bloque residual para el bloque de vídeo actual basándose en una diferencia entre el bloque de vídeo actual y el bloque de predicción para el bloque de vídeo actual, que puede ser la suma del bloque de vídeo de referencia y el bloque de predicción residual para el bloque de vídeo actual (344). El codificador de vídeo 20 puede codificar el flujo de bits de vídeo para identificar este bloque residual y el bloque de vídeo de referencia (346).

30 La FIG. 20 es un diagrama de flujo que ilustra un ejemplo de procedimiento de ARP de inter-vista para codificar un bloque de vídeo predicho de inter-vista de acuerdo con las técnicas descritas en esta divulgación. El procedimiento a modo de ejemplo de la FIG. 20 puede realizarse mediante un codificador de vídeo, tal como un codificador de vídeo 20, que puede incluir una unidad ARP 245.

35 De acuerdo con el procedimiento a modo de ejemplo de la FIG. 20, el codificador de vídeo 20, por ejemplo, la unidad ARP 245, identifica el DMV del bloque de vídeo actual (**Curr**) al bloque de vídeo de referencia de inter-vista (**Base**). El codificador de vídeo 20, por ejemplo, la unidad ARP 245, también identifica un TMV y una imagen de referencia asociada del bloque de vídeo de referencia de inter-vista (**Base**) (352). El codificador de vídeo 20, por ejemplo, la

40 unidad ARP 245, puede entonces identificar bloques de vídeo de referencia temporales en las vistas actuales y de referencia (**CurrTRef** y **BaseTRef**, respectivamente) basándose en el TMV, por ejemplo, utilizando las técnicas descritas anteriormente con respecto a la FIG. 10 (354).

45 El codificador de vídeo 30, por ejemplo, la unidad ARP 245, puede entonces determinar un bloque de predicción residual de inter-vista para el bloque de vídeo actual basándose en la diferencia entre estos bloques de vídeo de referencia temporales (**CurrTRef - BaseTRef**) (318). El codificador de vídeo 20, por ejemplo, la unidad ARP 245 y el sumador 250 pueden determinar un bloque residual para el bloque de vídeo actual basándose en una diferencia entre el bloque de vídeo actual y el bloque de predicción para el bloque de vídeo actual, que puede ser la suma del

50 bloque de vídeo de referencia de inter-vista (**Base**) y el bloque de predicción residual para el bloque de vídeo actual (358). El codificador de vídeo 20 puede codificar el flujo de bits de vídeo para identificar este bloque residual y el bloque de vídeo de referencia de inter-vista (360).

La FIG. 21 es un diagrama de flujo que ilustra un ejemplo de procedimiento de ARP temporal para codificar un bloque de vídeo predicho temporalmente de acuerdo con las técnicas descritas en esta divulgación. El

55 procedimiento a modo de ejemplo de la FIG. 21 puede realizarse mediante un codificador de vídeo, tal como un codificador de vídeo 20, que puede incluir una unidad ARP 245.

De acuerdo con el procedimiento a modo de ejemplo de la FIG. 21, el codificador de vídeo 20, por ejemplo la unidad ARP 245, identifica un bloque de vídeo de referencia temporal (**CurrTRef**) en la vista actual, por ejemplo, usando un

60 TMV asociado con el bloque de vídeo actual. El codificador de vídeo 20, por ejemplo, la unidad ARP 245, puede entonces identificar un DMV del bloque de vídeo de referencia temporal (**CurrTRef**), que a su vez puede identificar un bloque de vídeo de referencia temporal en la vista de referencia (**BaseTRef**) (370). Basándose en el DMV del bloque de vídeo de referencia temporal (**CurrTRef**) en la vista actual, el codificador de vídeo 20, por ejemplo, la unidad ARP 245, también puede identificar un bloque de vídeo de referencia de inter-vista en la vista de referencia

65 (**Base**) (372).

- El codificador de vídeo 20, por ejemplo, la unidad ARP 245, puede determinar entonces el bloque de predicción residual temporal para el bloque de vídeo actual basándose en la diferencia entre estos bloques de vídeo de referencia en la vista de referencia (**Base - BaseTRef**) (374). El codificador de vídeo 20, por ejemplo la unidad ARP 245 y el sumador 250, pueden determinar un bloque residual para el bloque de vídeo actual basándose en una
- 5 diferencia entre el bloque de vídeo actual y el bloque de predicción para el bloque de vídeo actual, que puede ser la suma del bloque de vídeo de referencia temporal (**CurrTRef**) y el bloque de predicción residual para el bloque de vídeo actual (376). El codificador de vídeo 20 puede codificar el flujo de bits de vídeo para identificar este bloque residual y el bloque de vídeo de referencia de inter-vista (378).
- 10 La FIG. 22 es un diagrama de flujo que ilustra un procedimiento de ejemplo para identificar un DMV para ARP temporal de acuerdo con las técnicas descritas en esta divulgación. El procedimiento a modo de ejemplo de la FIG. 22 puede realizarse mediante un codificador de vídeo, tal como el codificador de vídeo 20 y/o el descodificador de vídeo 30, que puede incluir una unidad ARP 245, 283.
- 15 De acuerdo con el procedimiento a modo de ejemplo de la FIG. 22, el codificador de vídeo identifica un bloque de vídeo de referencia temporal en la vista actual (**CurrTRef**) basándose en un TMV escalado (380). El codificador de vídeo determina entonces el bloque de vídeo de referencia temporal identificado que está asociado con un DMV (382). Si el bloque de vídeo de referencia temporal está asociado con un DMV, el codificador de vídeo identifica un bloque de vídeo de referencia de inter-vista basándose en el DMV (388). Si el bloque de vídeo de referencia
- 20 temporal no está asociado con un DMV, el codificador de vídeo identifica otro bloque de vídeo de referencia temporal en la vista actual basándose en el escalado ausente de TMV (384) e identifica el bloque de vídeo de referencia de inter-vista basándose en un DMV del bloque de vídeo de referencia temporal en la vista actual que se identificó basándose en el escalado ausente de TMV (388).
- 25 La FIG. 23 es un diagrama de flujo que ilustra un procedimiento de ejemplo para identificar un DMV o TMV para ARP de acuerdo con las técnicas descritas en esta divulgación. El procedimiento a modo de ejemplo de la FIG. 23 puede realizarse mediante un codificador de vídeo, tal como el codificador de vídeo 20 y/o el descodificador de vídeo 30, que puede incluir una unidad ARP 245, 283.
- 30 De acuerdo con el procedimiento a modo de ejemplo de la FIG. 23, el codificador de vídeo comprueba primero RefPicList0 para el DMV o TMV necesario para ARP (390). Si RefPicList0 incluye el DMV o TMV, el codificador de vídeo identifica un bloque de vídeo de referencia basándose en el DMV o TMV (396). Si RefPicList0 no incluye el DMV o TMV, el codificador de vídeo comprueba RefPicList1 para el DMV o TMV (394), y puede identificar un bloque de vídeo de referencia basándose en el DMV o TMV de RefPicList1 (396). Si ninguna de las listas de imágenes de
- 35 referencia incluye el DMV o el TMV, el codificador de vídeo puede utilizar un vector de movimiento cero, o no realizar ARP, como ejemplos. En algunos ejemplos en los que el codificador de vídeo utiliza un vector de movimiento cero para ARP, el codificador de vídeo puede aplicar el vector de movimiento cero a la lista de imágenes de referencia (dirección) invocada para la predicción de inter-vista utilizando el DMV.
- 40 En algunos ejemplos, uno o más aspectos de las técnicas descritas en esta divulgación pueden ser realizados por un dispositivo de red intermedio, tal como un elemento de red consciente de medios (MANE), un procesador de adaptación de flujo, un procesador de empalme o un procesador de edición. Por ejemplo, dicho dispositivo intermedio puede configurarse para generar o recibir cualquiera de una variedad de señales como se describe en esta divulgación.
- 45 En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, firmware o cualquier combinación de estos. Si se implementan en software, las funciones, como una o más instrucciones o código, pueden almacenarse en, o transmitirse por, un medio legible por ordenador, y ejecutarse mediante una unidad de procesamiento basándose en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible tal como unos medios de almacenamiento de datos, o medios de comunicación que incluyen cualquier medio que facilite la transferencia de un programa informático de un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador pueden corresponder, en general, a (1) medios de almacenamiento tangibles y legibles por ordenador, que sean no transitorios, o (2) un medio de comunicación tal como una señal o
- 50 una onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que se puede acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para implementar las técnicas descritas en la presente divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.
- 55 A modo de ejemplo, y no de manera limitativa, dichos 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 utilizarse 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 puede denominarse de forma correcta con el
- 60 término medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, un servidor u otra fuente remota mediante un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado
- 65

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. Los discos, como se usan en el presente documento, incluyen el disco compacto (CD), el disco de láser, el disco óptico, el disco versátil digital (DVD), el disco flexible y el disco Blu-ray, donde algunos discos normalmente reproducen datos de manera magnética, mientras que otros discos reproducen los datos de manera óptica con láser. Las combinaciones de lo anterior también deberían incluirse dentro del alcance de los medios legibles por ordenador.

Las instrucciones pueden ser ejecutadas por uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), matrices lógicas programables in situ (FPGA) u otros circuitos lógicos integrados o discretos equivalentes. Por consiguiente, el término «procesador», como se utiliza en el presente documento, puede referirse a cualquier estructura anterior 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 puede proporcionarse dentro de módulos de hardware y/o software específicos configurados para la codificación y la decodificación, o incorporarse en un códec combinado. Además, las técnicas podrían implementarse completamente en uno o más circuitos o elementos lógicos.

Las técnicas de la presente divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, que incluyen un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En la presente divulgación se describen varios componentes, módulos o unidades para enfatizar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no requieren necesariamente su realización mediante diferentes unidades de hardware. En su lugar, como se ha descrito anteriormente, diversas unidades pueden combinarse en una unidad de hardware de códec o proporcionarse por medio de un grupo de unidades de hardware interoperativas, que incluyen uno o más procesadores como los descritos anteriormente, conjuntamente con software y/o firmware adecuado. Se han descrito diversos ejemplos.

REIVINDICACIONES

1. Un procedimiento de predicción residual de inter-vista para descodificar datos de vídeo HEVC, comprendiendo el procedimiento:

descodificar un flujo de bits de vídeo codificado que codifica los datos de vídeo para identificar un vector de movimiento de disparidad (DMV) y un bloque residual para un bloque de vídeo actual, en el que el bloque de vídeo actual está en una vista actual y en el que el DMV se utiliza para predicción de inter-vista del bloque de vídeo actual basándose en un bloque de vídeo de referencia de inter-vista en una vista de referencia y en una misma unidad de acceso que el bloque de vídeo actual;

identificar un vector de movimiento temporal (TMV) y una imagen de referencia asociada del bloque de vídeo de referencia de inter-vista;

identificar un bloque de vídeo de referencia temporal en la imagen de referencia asociada en la vista de referencia basándose en el TMV del bloque de vídeo de referencia de inter-vista;

identificar un bloque de vídeo de referencia temporal en la vista actual basándose en el TMV del bloque de vídeo de referencia de inter-vista en la vista de referencia, en el que el bloque de vídeo de referencia temporal en la vista actual y el bloque de vídeo de referencia temporal en la vista de referencia se encuentran en la misma unidad de acceso;

determinar un bloque de predicción residual para el bloque de vídeo actual basándose en una diferencia entre el bloque de vídeo de referencia temporal en la vista actual y el bloque de vídeo de referencia temporal en la vista de referencia; y

aplicar el bloque de predicción residual y el bloque residual identificado desde el flujo de bits de vídeo codificado al bloque de vídeo de referencia para reconstruir el bloque de vídeo actual.

2. El procedimiento según la reivindicación 1, en el que la determinación del bloque de predicción residual para el bloque de vídeo actual basándose en la diferencia entre el bloque de vídeo de referencia temporal en la vista actual y el bloque de vídeo de referencia temporal en la vista de referencia comprende aplicar un factor de ponderación a la diferencia entre el bloque de vídeo de referencia temporal en la vista actual y el bloque de vídeo de referencia temporal en la vista de referencia.

3. El procedimiento según la reivindicación 1, que comprende además escalar el TMV del bloque de vídeo de referencia de inter-vista a una imagen de referencia objetivo en una unidad de acceso objetivo para la predicción residual del bloque de vídeo actual, en el que al menos uno de:

identificar el bloque de vídeo de referencia temporal en la vista de referencia basándose en el TMV del bloque de vídeo de referencia de inter-vista comprende identificar el bloque de vídeo de referencia temporal en la vista de referencia basándose en el TMV escalado; e

identificar el bloque de vídeo de referencia temporal en la vista actual basándose en el TMV del bloque de vídeo de referencia de inter-vista comprende identificar el bloque de vídeo de referencia temporal en la vista de referencia basándose en el TMV escalado.

4. El procedimiento según la reivindicación 3, que comprende además:

seleccionar una lista de imágenes de referencia; y

seleccionar la imagen de referencia objetivo a partir de la lista de imágenes de referencia seleccionada, en el que seleccionar la lista de imágenes de referencia comprende una de las siguientes:

seleccionar la lista de imágenes de referencia asociada con el TMV; o

seleccionar la lista de imágenes de referencia asociada con el DMV.

5. El procedimiento según la reivindicación 1, en el que el bloque de vídeo de referencia de inter-vista comprende una pluralidad de unidades de predicción, e identificar el TMV del bloque de vídeo de referencia de inter-vista comprende identificar un TMV asociado con una de la pluralidad de unidades de predicción que contiene una posición central del bloque de vídeo de referencia de inter-vista.

6. El procedimiento según la reivindicación 1, en el que el bloque de vídeo de referencia de inter-vista contiene un primer conjunto de información de movimiento correspondiente a una primera lista de imágenes de referencia y un segundo conjunto de información de movimiento correspondiente a una segunda lista de

imágenes de referencia, e identificar el TMV del bloque de vídeo de referencia de inter-vista comprende:

seleccionar el TMV del primer conjunto de información de movimiento si el primer conjunto de información de movimiento incluye un TMV; y

seleccionar el TMV del segundo conjunto de información de movimiento si el primer conjunto de información de movimiento no incluye un TMV.

7. El procedimiento según la reivindicación 1, que comprende además:

aplicar escalado de diferencia de orden de vista para escalar el DMV identificado a una vista de referencia objetivo para la predicción residual del bloque de vídeo actual; e

identificar el bloque de vídeo de referencia de inter-vista basándose en el DMV escalado.

8. Un procedimiento de predicción residual de inter-vista para codificar datos de vídeo HEVC, comprendiendo el procedimiento:

identificar un vector de movimiento de disparidad (DMV) para un bloque de vídeo actual, en el que el bloque de vídeo actual está en una vista actual y en el que el DMV se utiliza para la predicción de inter-vista del bloque de vídeo actual basándose en un bloque de vídeo de referencia de inter-vista en una vista de referencia y en la misma unidad de acceso que el bloque de vídeo actual;

identificar un vector de movimiento temporal (TMV) y una imagen de referencia asociada del bloque de vídeo de referencia de inter-vista;

identificar un bloque de vídeo de referencia temporal en la imagen de referencia asociada en la vista de referencia basándose en el TMV del bloque de vídeo de referencia de inter-vista;

identificar un bloque de vídeo de referencia temporal en la vista actual basándose en el TMV del bloque de vídeo de referencia de inter-vista en la vista de referencia, en el que el bloque de vídeo de referencia temporal en la vista actual y el bloque de vídeo de referencia temporal en la vista de referencia se encuentran en la misma unidad de acceso;

determinar un bloque de predicción residual para el bloque de vídeo actual basándose en una diferencia entre el bloque de vídeo de referencia temporal en la vista actual y el bloque de vídeo de referencia temporal en la vista de referencia; y

codificar un flujo de bits de vídeo codificado que codifica los datos de vídeo para identificar el DMV y un bloque residual para el bloque de vídeo actual, en el que el bloque residual identificado por el flujo de bits de vídeo codificado comprende una diferencia entre el bloque de vídeo de referencia de inter-vista y el bloque de predicción residual para el bloque de vídeo actual.

9. El procedimiento según la reivindicación 8, en el que la determinación del bloque de predicción residual para el bloque de vídeo actual basándose en la diferencia entre el bloque de vídeo de referencia temporal en la vista actual y el bloque de vídeo de referencia temporal en la vista de referencia comprende aplicar un factor de ponderación a la diferencia entre el bloque de vídeo de referencia temporal en la vista actual y el bloque de vídeo de referencia temporal en la vista de referencia.

10. El procedimiento según la reivindicación 8, que comprende además escalar el TMV del bloque de vídeo de referencia de inter-vista a una imagen de referencia objetivo en una unidad de acceso objetivo para la predicción residual del bloque de vídeo actual, en el que al menos uno de:

identificar el bloque de vídeo de referencia temporal en la vista de referencia basándose en el TMV del bloque de vídeo de referencia de inter-vista comprende identificar el bloque de vídeo de referencia temporal en la vista de referencia basándose en el TMV escalado; y

identificar el bloque de vídeo de referencia temporal en la vista actual basándose en el TMV del bloque de vídeo de referencia de inter-vista comprende identificar el bloque de vídeo de referencia temporal en la vista de referencia basándose en el TMV escalado.

11. El procedimiento según la reivindicación 10, que comprende además:

seleccionar una lista de imágenes de referencia; y

seleccionar la imagen de referencia objetivo de la lista de imágenes de referencia seleccionada,

en el que la selección de la lista de imágenes de referencia comprende una de las siguientes:

seleccionar la lista de imágenes de referencia asociada con el TMV; o

5 seleccionar la lista de imágenes de referencia asociada con el DMV.

12. Un aparato que comprende un codificador de vídeo configurado para realizar una predicción residual de inter-vista para codificar datos de vídeo HEVC, comprendiendo el codificador de vídeo:

10 una memoria configurada para almacenar un flujo de bits de vídeo codificado que codifica los datos de vídeo; y

uno o más procesadores configurados para:

15 identificar un vector de movimiento de disparidad (DMV) para un bloque de vídeo actual, en el que el bloque de vídeo actual está en una vista actual y en el que el DMV se utiliza para la predicción de inter-vista del bloque de vídeo actual basándose en un bloque de vídeo de referencia de inter-vista en una vista de referencia y en la misma unidad de acceso que el bloque de vídeo actual;

20 identificar un vector de movimiento temporal (TMV) y una imagen de referencia asociada del bloque de vídeo de referencia de inter-vista;

identificar un bloque de vídeo de referencia temporal en la imagen de referencia asociada en la vista de referencia basándose en el TMV del bloque de vídeo de referencia de inter-vista;

25 identificar un bloque de vídeo de referencia temporal en la vista actual basándose en el TMV del bloque de vídeo de referencia de inter-vista en la vista de referencia, en el que el bloque de vídeo de referencia temporal en la vista actual y el bloque de vídeo de referencia temporal en la vista de referencia se encuentran en la misma unidad de acceso;

30 determinar un bloque de predicción residual para el bloque de vídeo actual basándose en una diferencia entre el bloque de vídeo de referencia temporal en la vista actual y el bloque de vídeo de referencia temporal en la vista de referencia; y codificar el flujo de bits de vídeo codificado para identificar el DMV y un bloque residual para el bloque de vídeo actual, en el que el bloque residual identificado mediante la codificación del flujo de bits de vídeo codificado comprende una diferencia entre el bloque de vídeo de referencia de inter-vista y el bloque de predicción residual para el bloque de vídeo actual.

40 **13.** El aparato de la reivindicación 12, en el que el codificador de vídeo es un decodificador de vídeo, y el uno o más procesadores están configurados para:

45 decodificar el flujo de bits de vídeo codificado para identificar el DMV y el bloque residual para el bloque de vídeo actual; y aplicar el bloque de predicción residual y el bloque residual identificado desde el flujo de bits de vídeo codificado al bloque de vídeo de referencia de inter-vista para reconstruir el bloque de vídeo actual.

14. El aparato de la reivindicación 12, en el que el codificador de vídeo es un codificador de vídeo y el uno o más procesadores están configurados para codificar el flujo de bits de vídeo codificado para indicar el DMV y el bloque residual para el bloque de vídeo actual a un decodificador de vídeo.

50 **15.** Un medio de almacenamiento legible por ordenador que presenta instrucciones almacenadas en el mismo que, al ejecutarse, hacen que uno o más procesadores de un codificador de vídeo emprendan el procedimiento según cualquiera de las reivindicaciones 1 a 11.

55

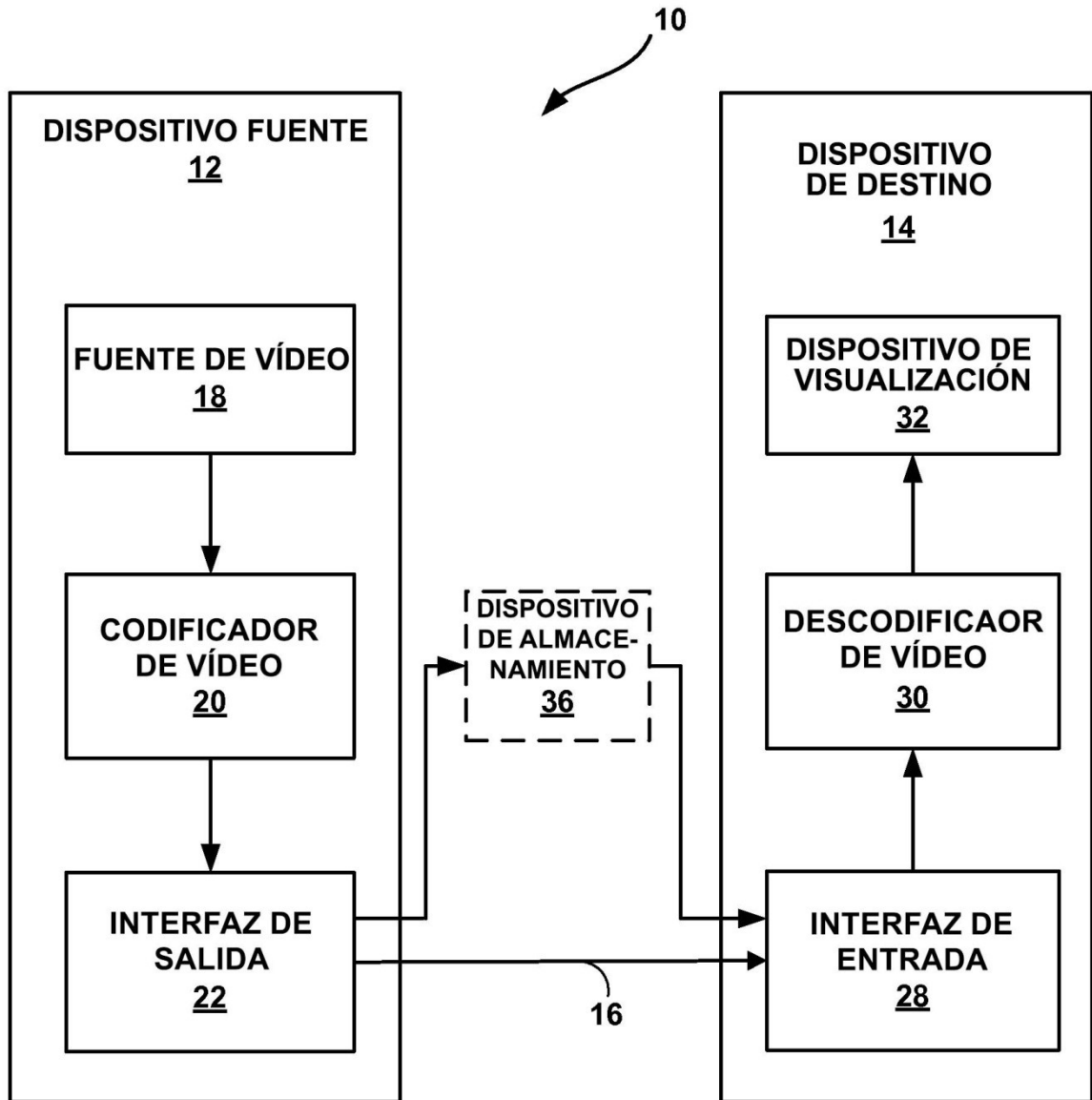


FIG. 1

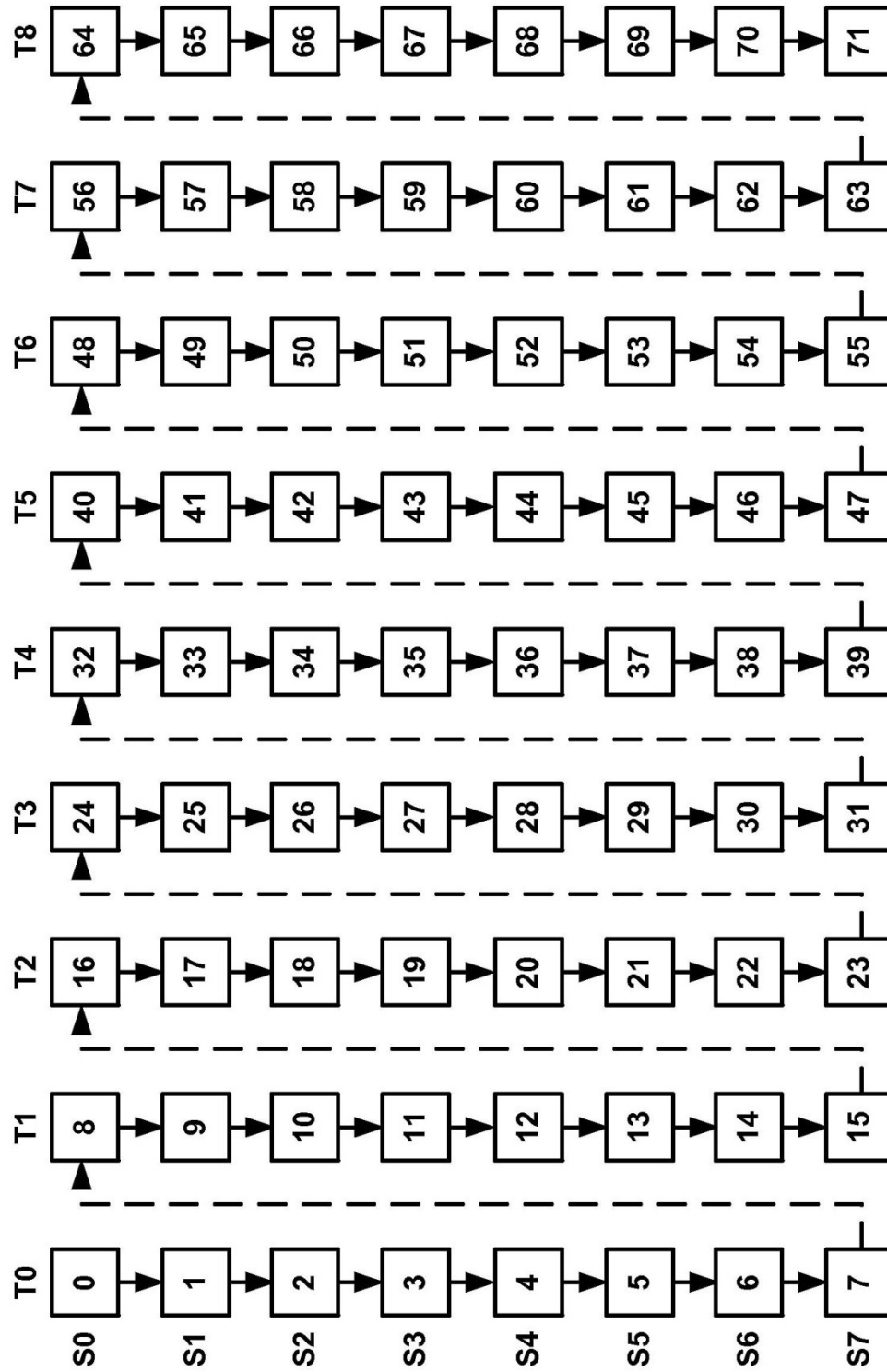


FIG. 2

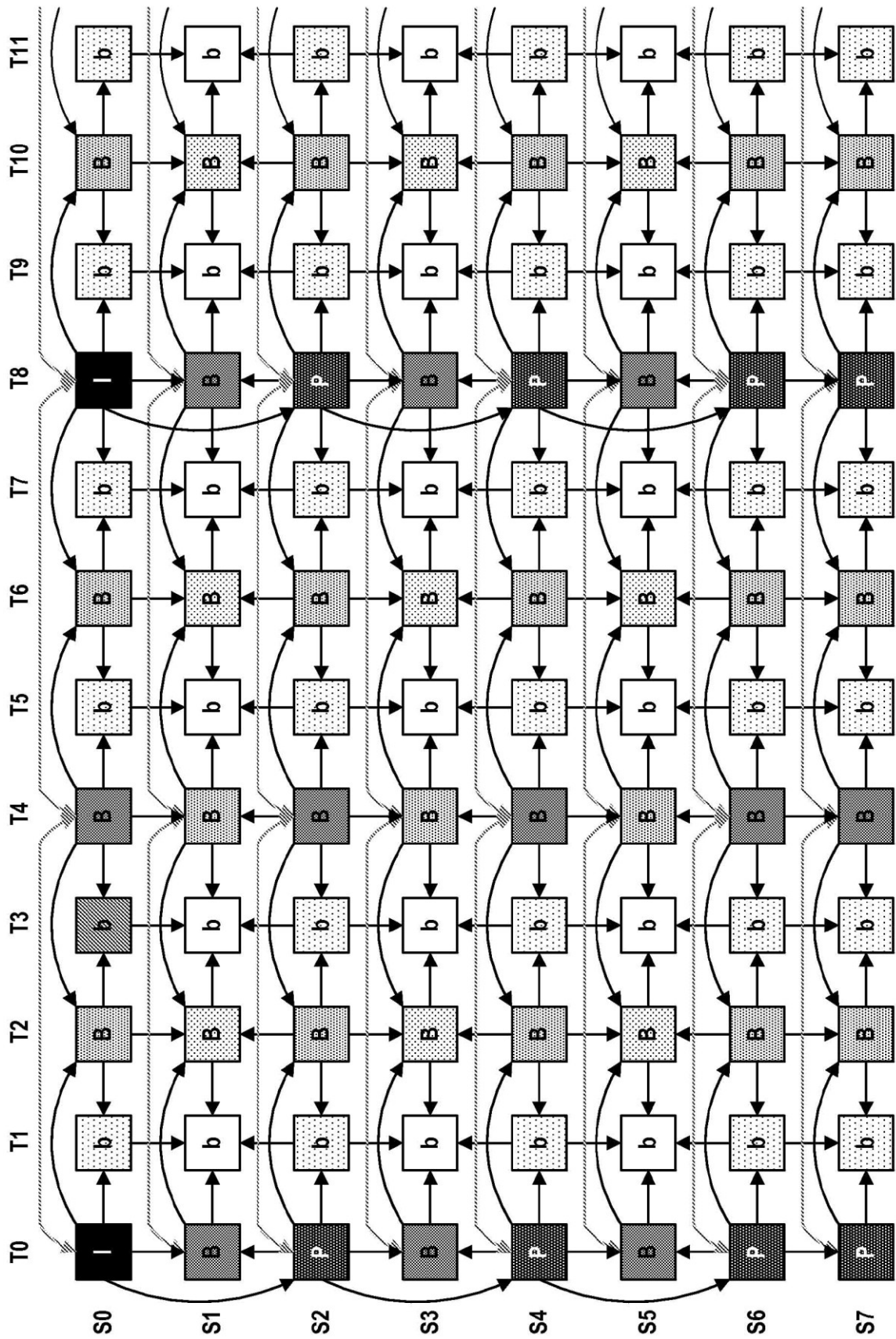


FIG. 3

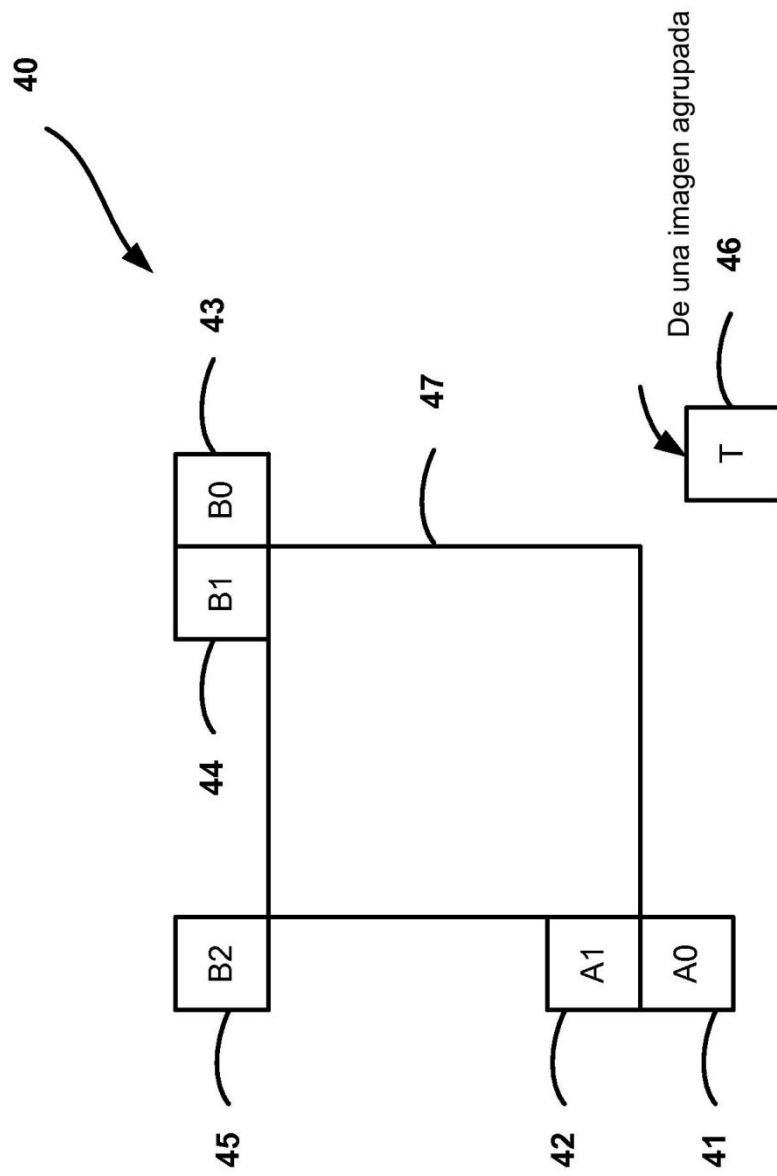


FIG. 4

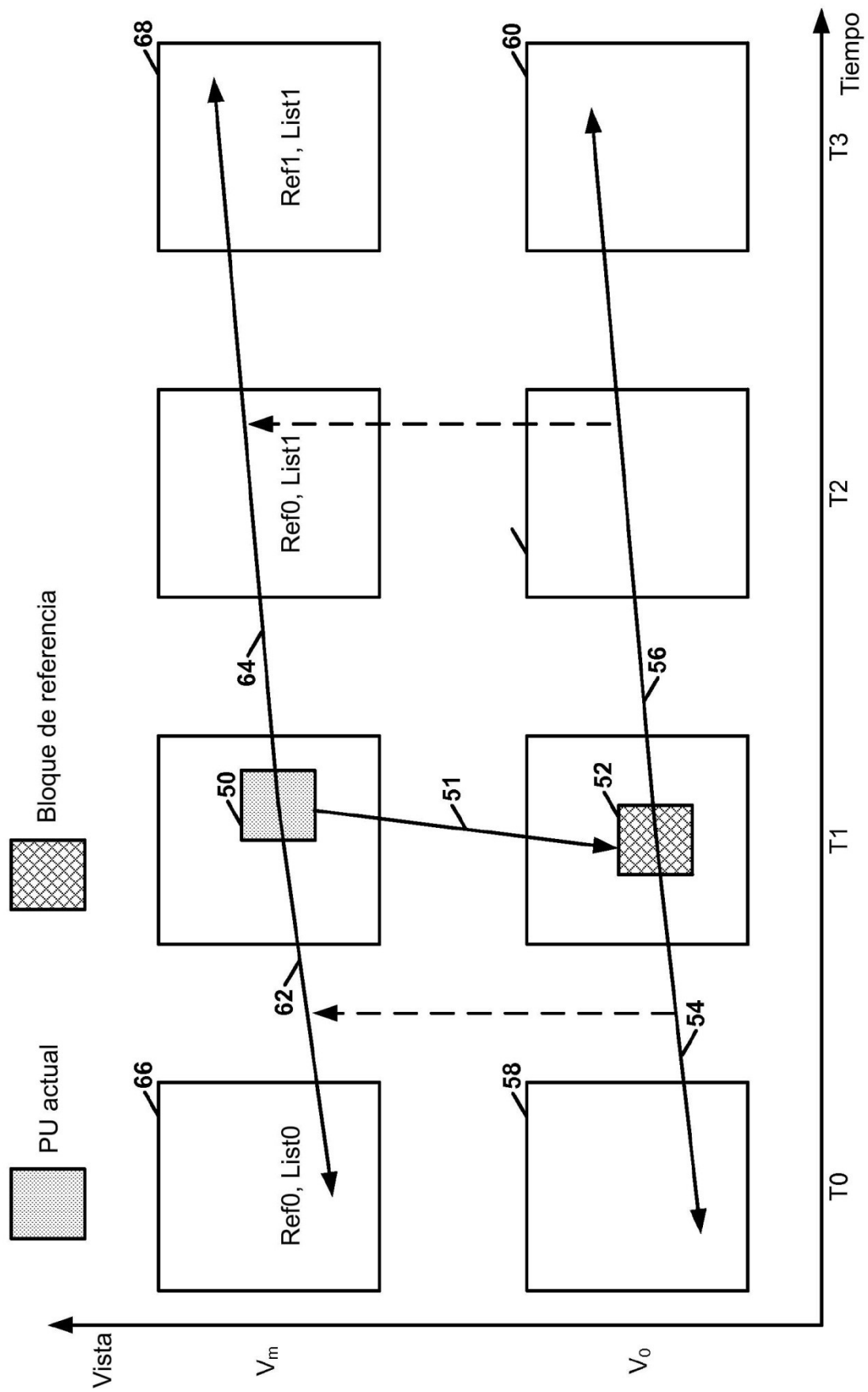


FIG. 5

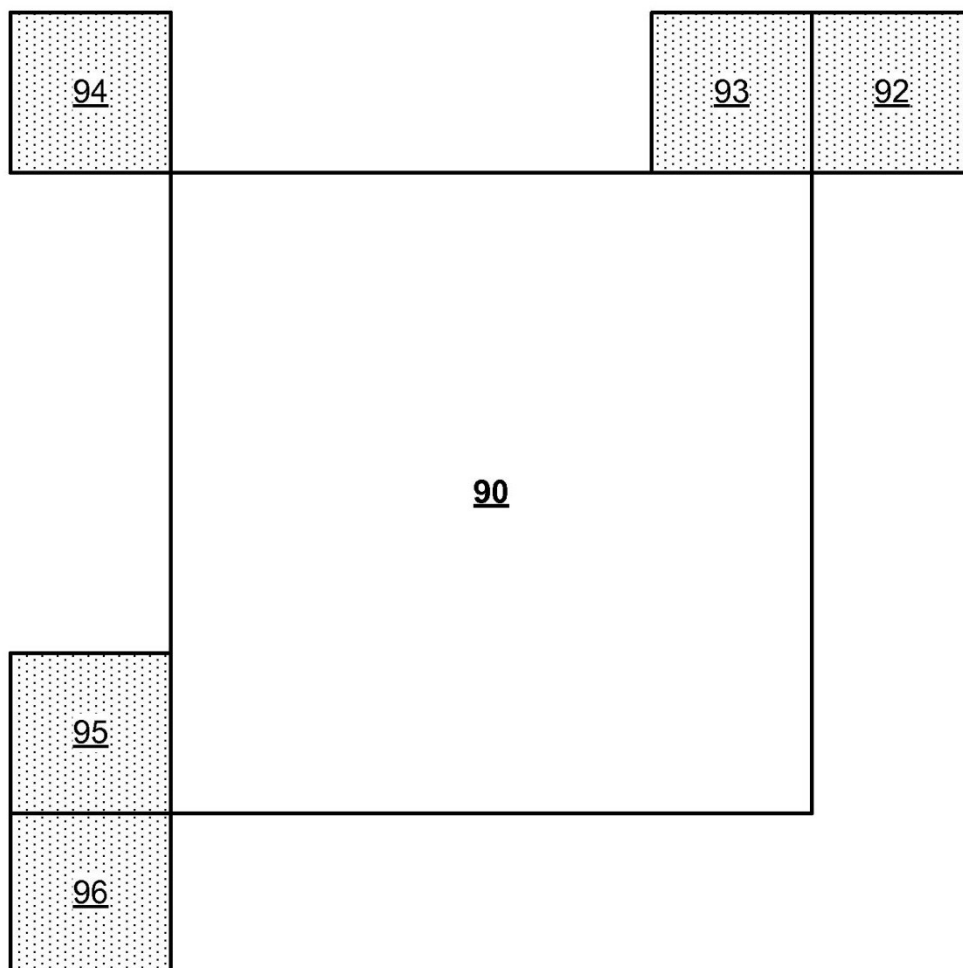


FIG. 6

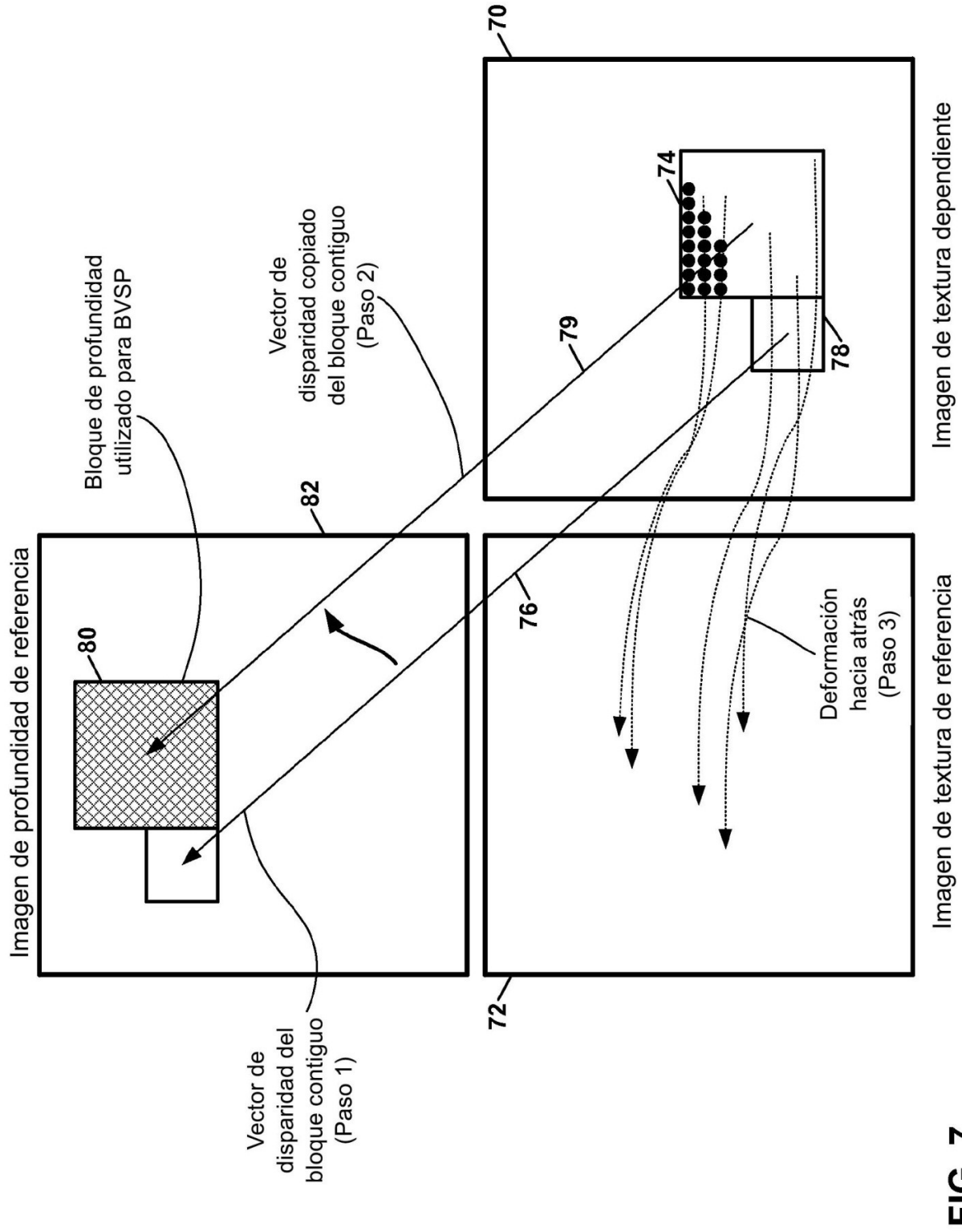


FIG. 7

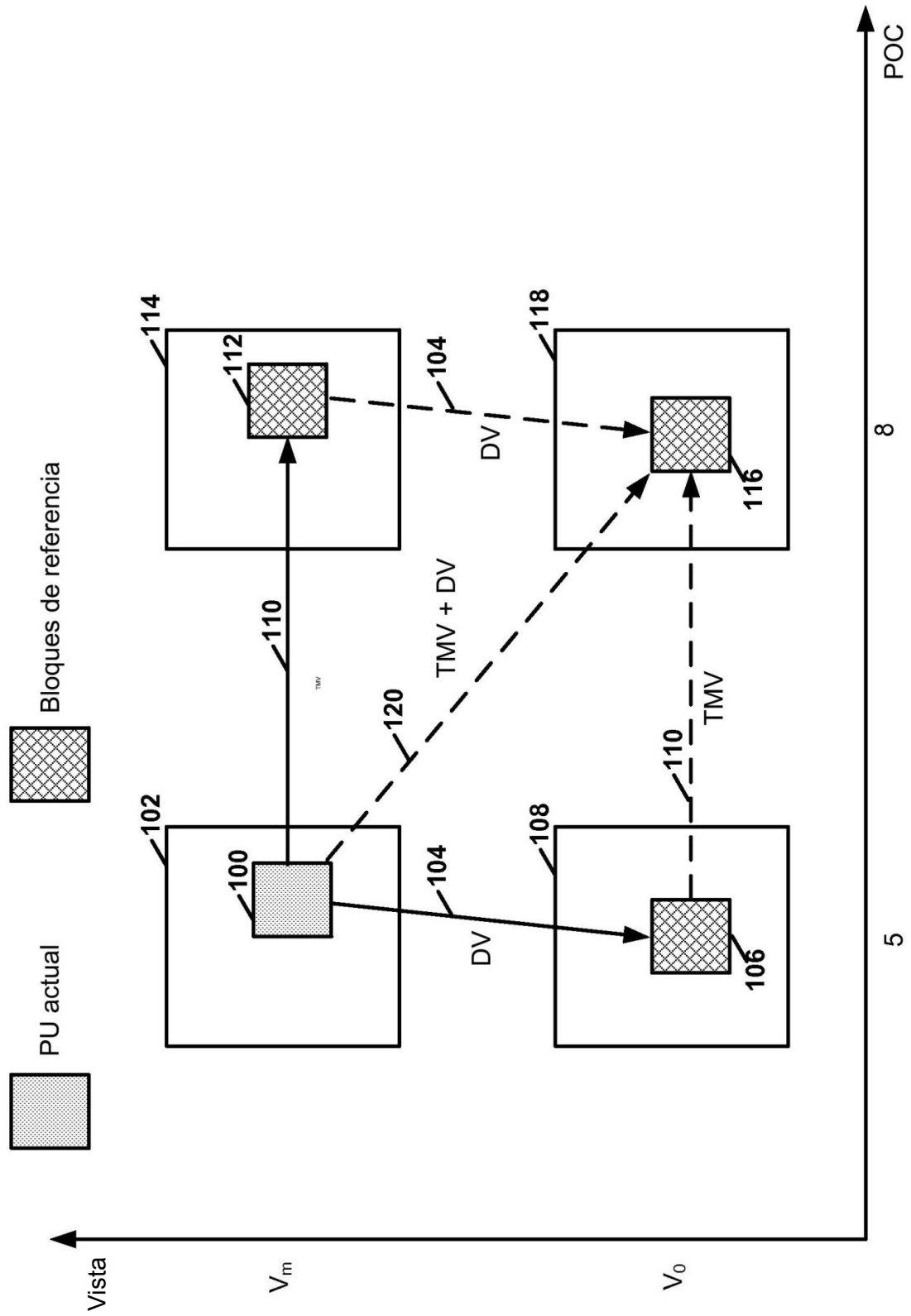


FIG. 8

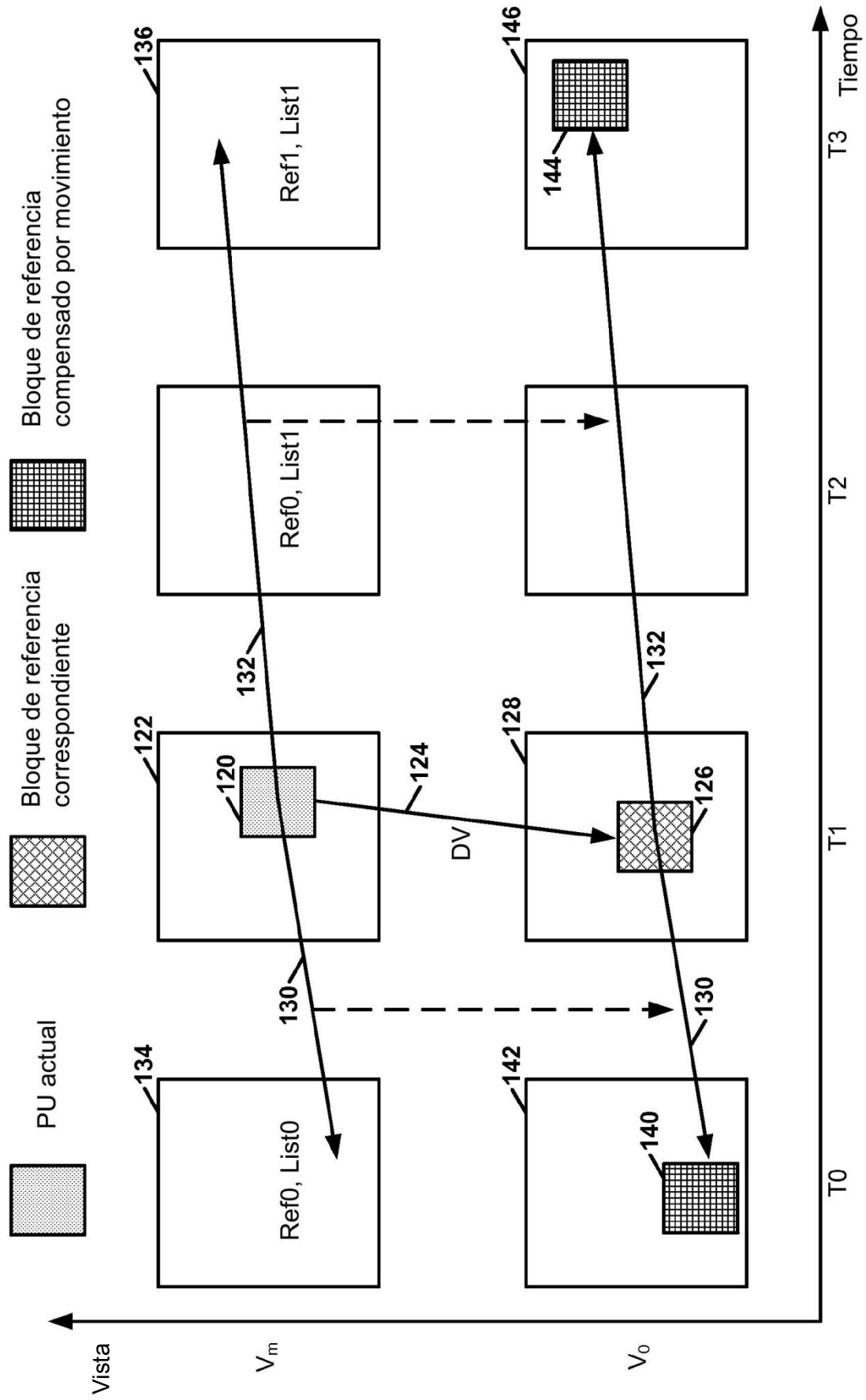


FIG. 9

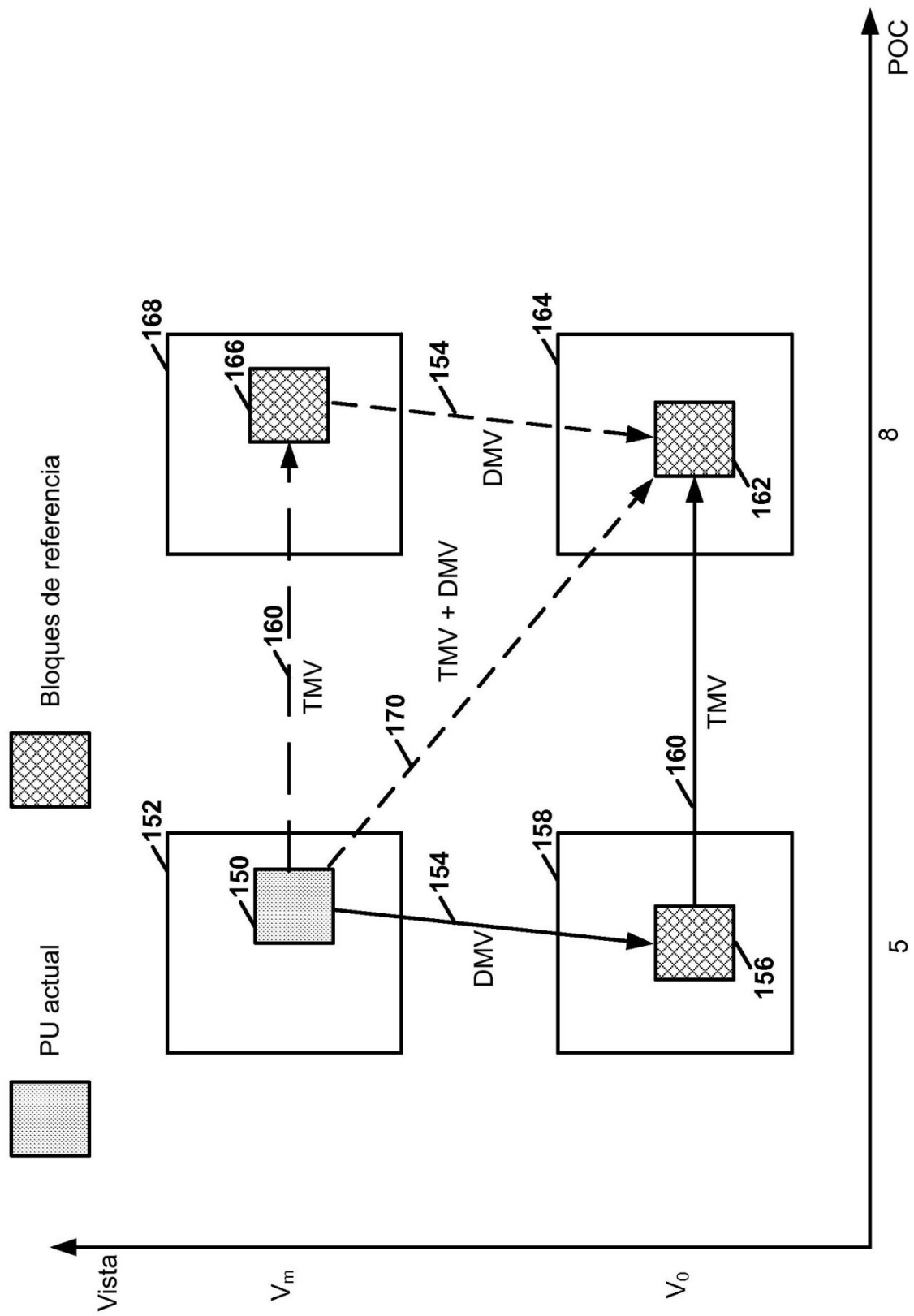


FIG. 10

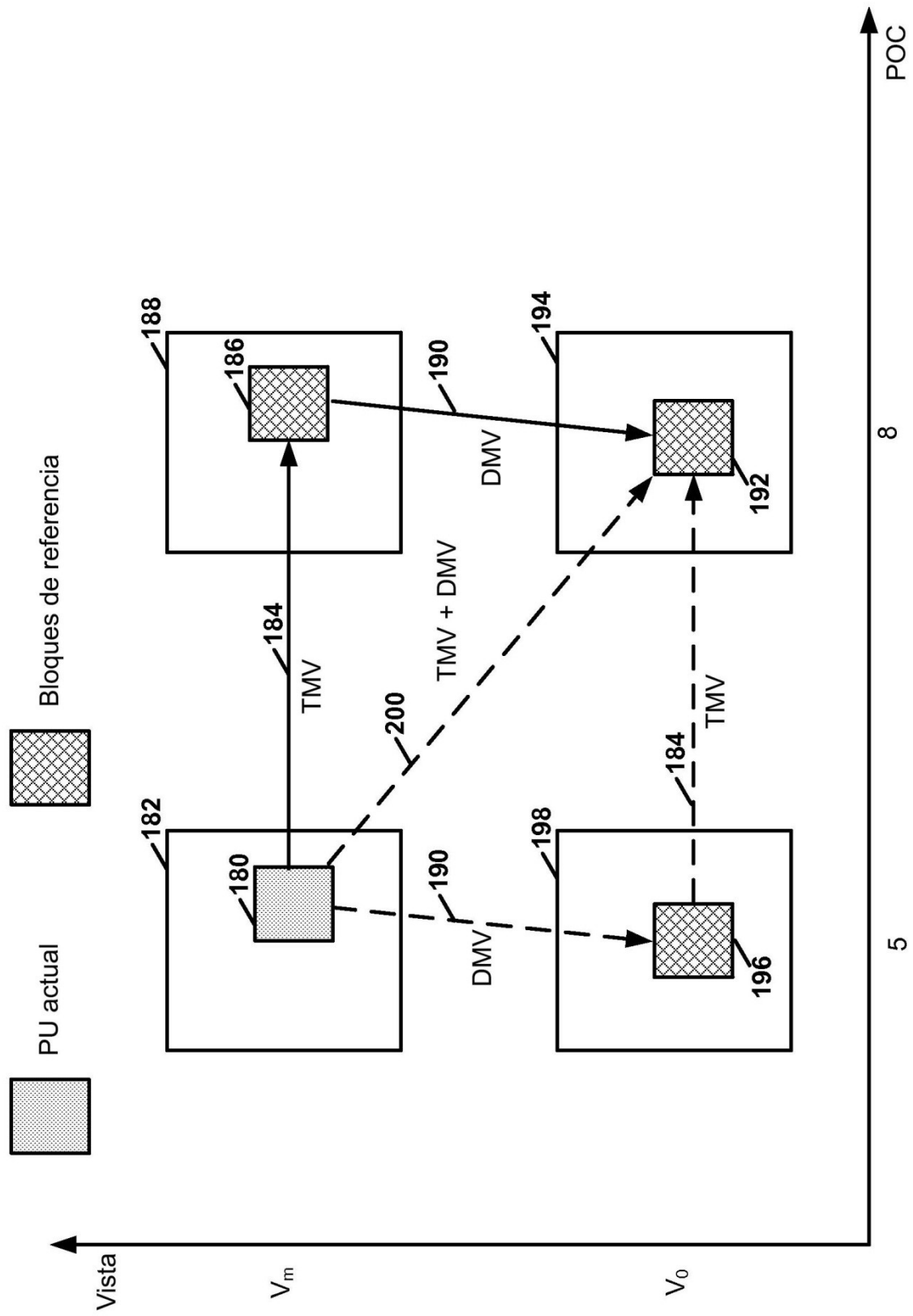


FIG. 11

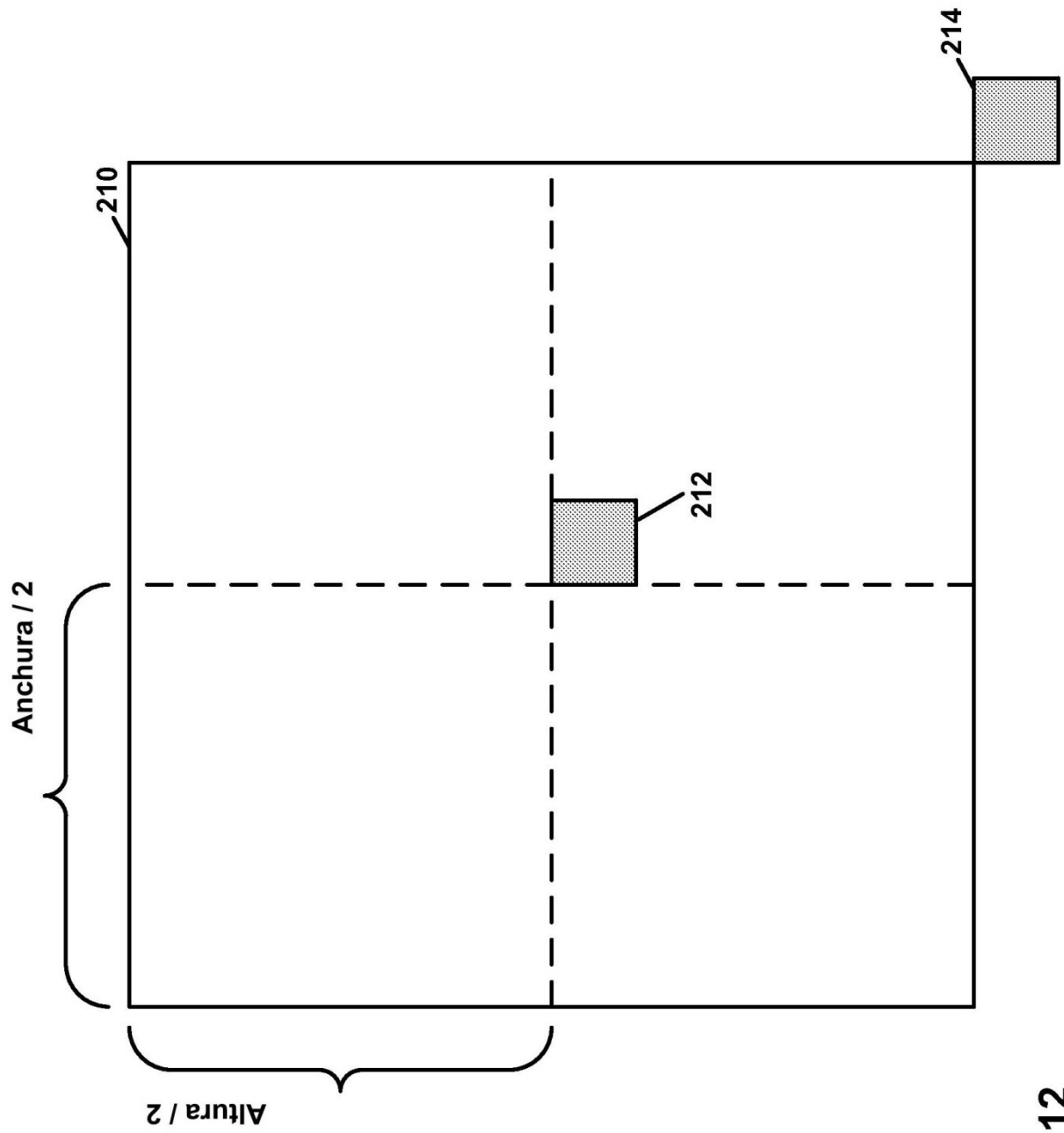


FIG. 12

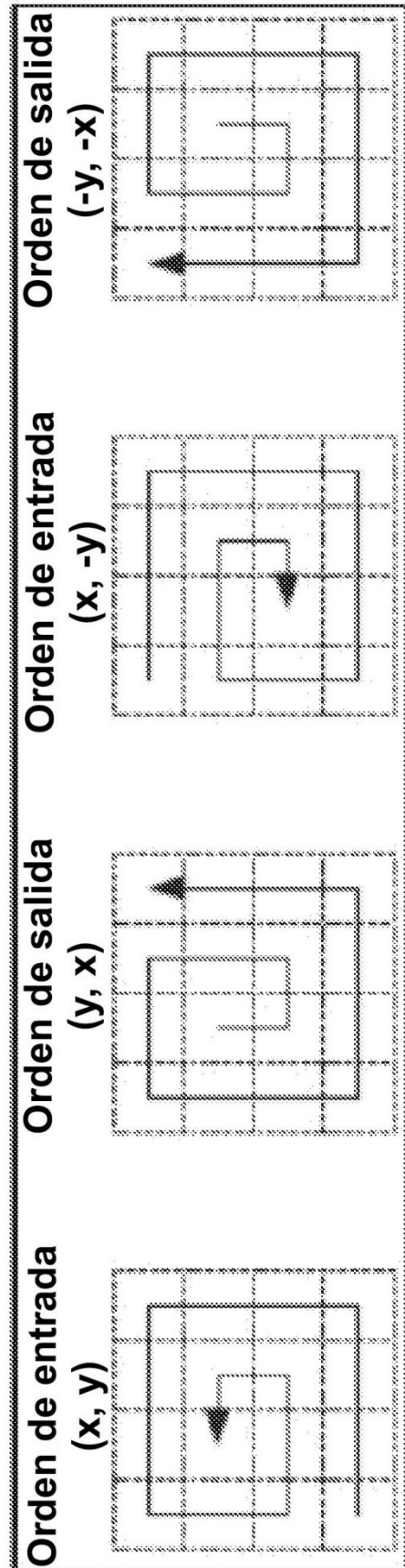


FIG. 13A

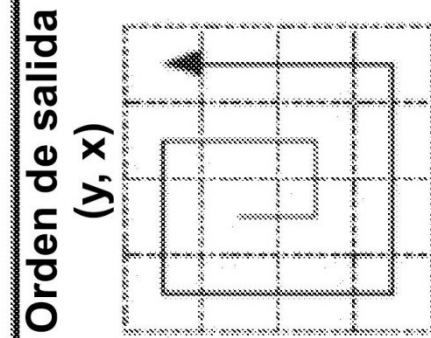


FIG. 13B

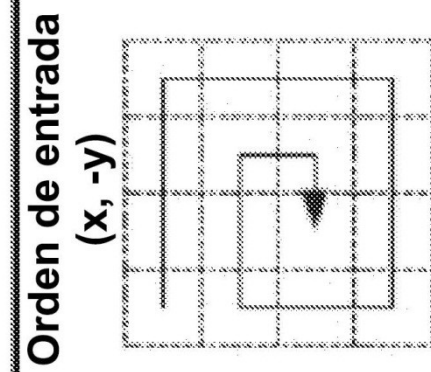


FIG. 13C

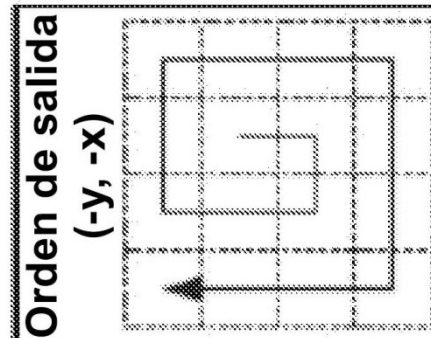


FIG. 13D

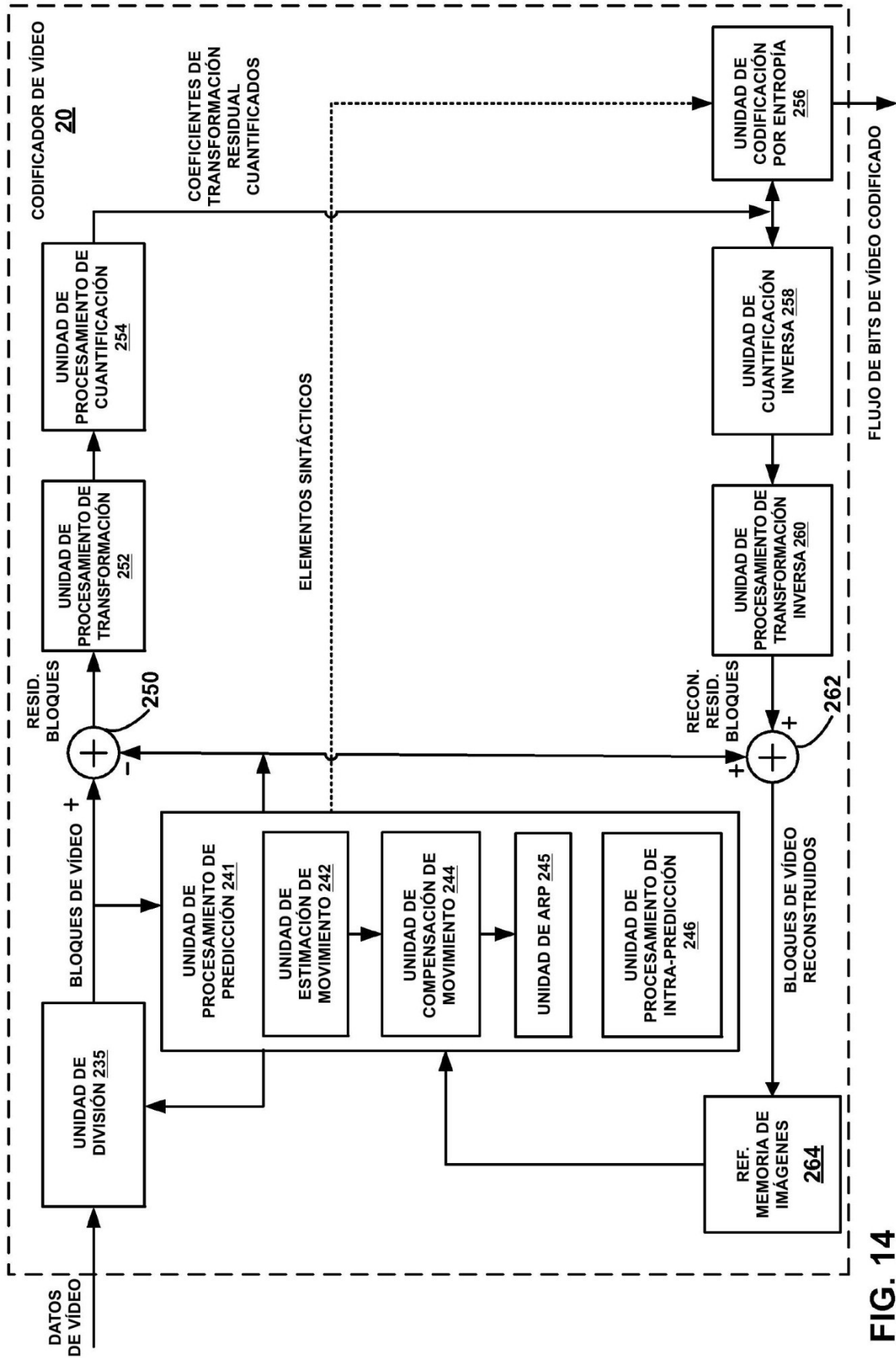


FIG. 14

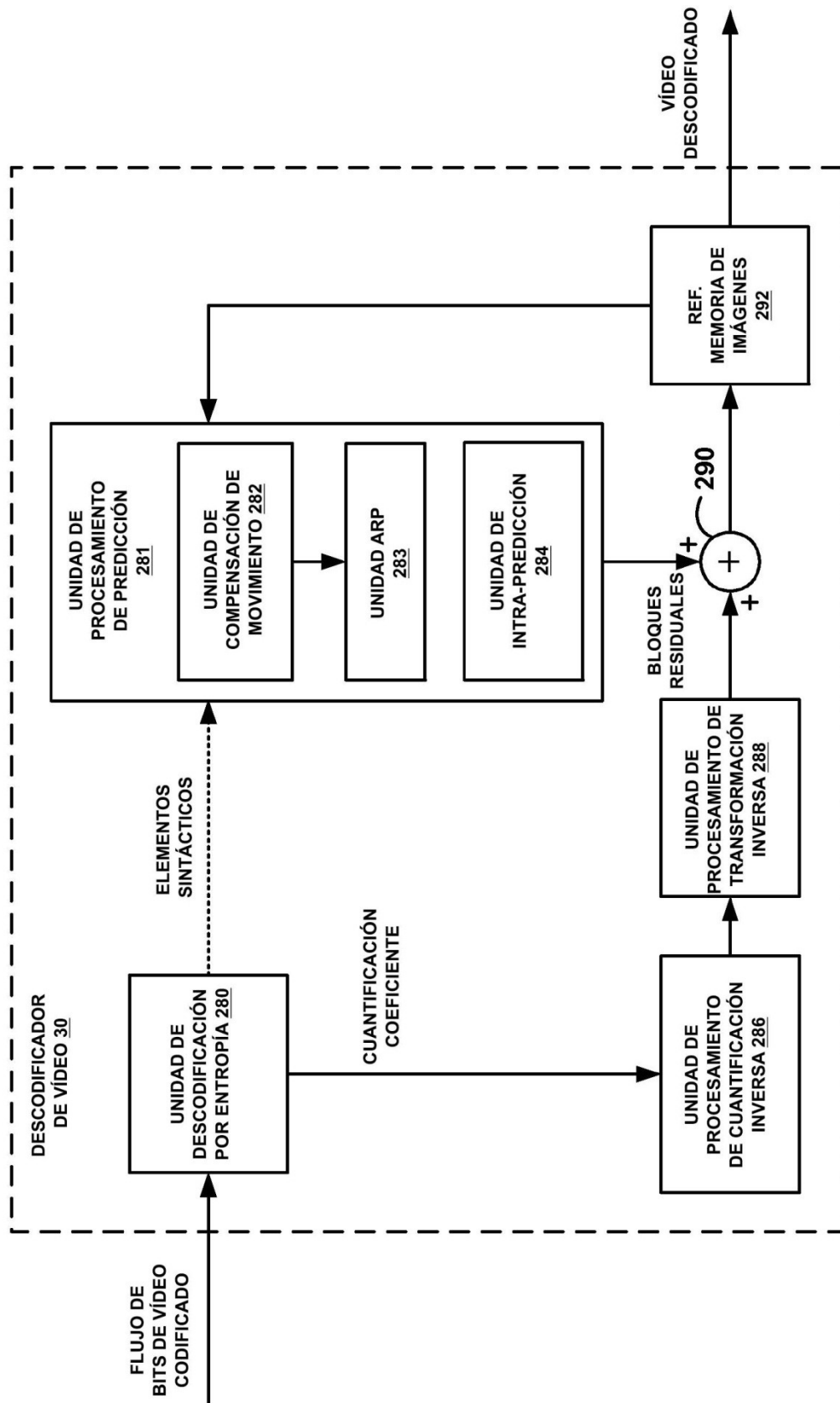


FIG. 15

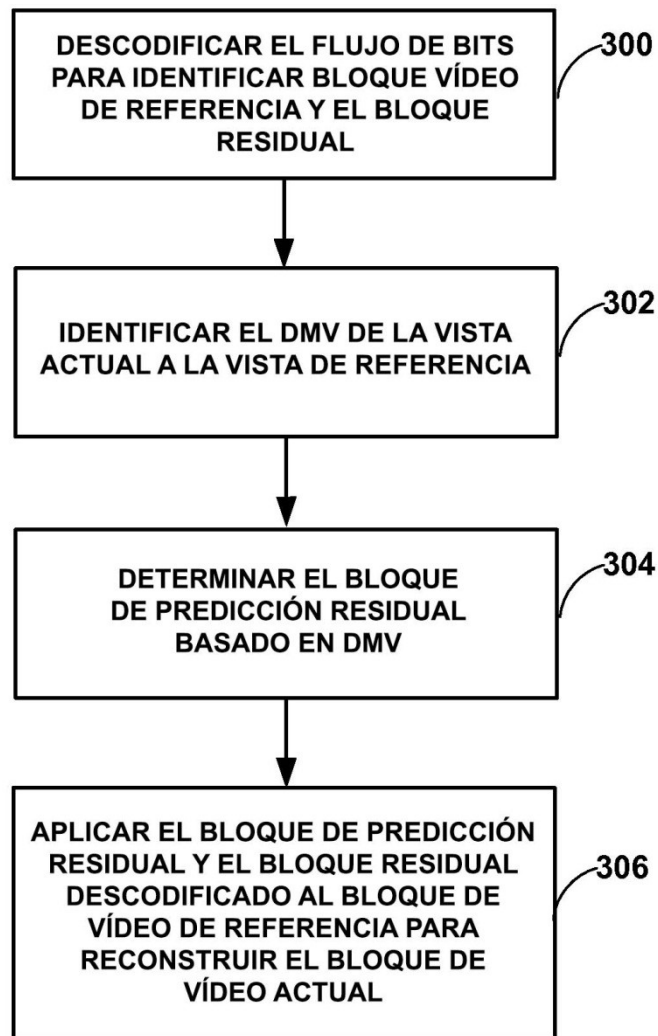


FIG. 16

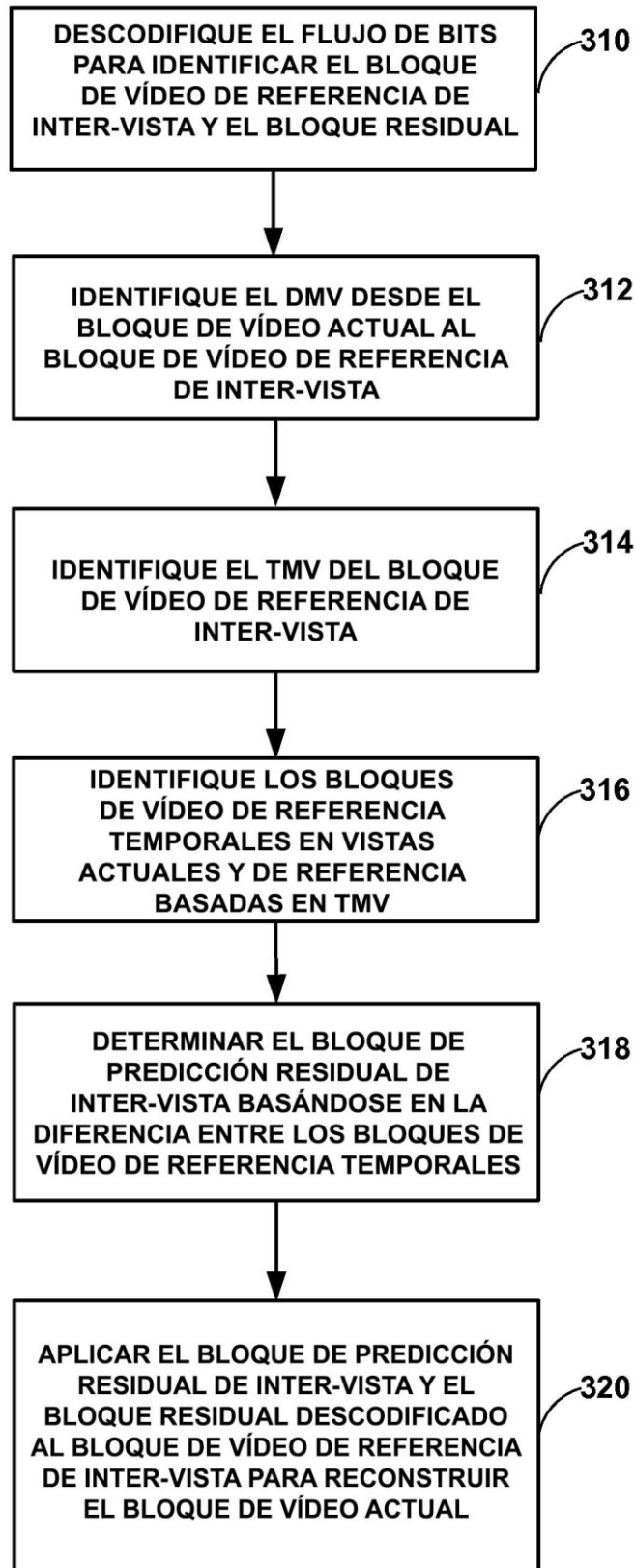


FIG. 17

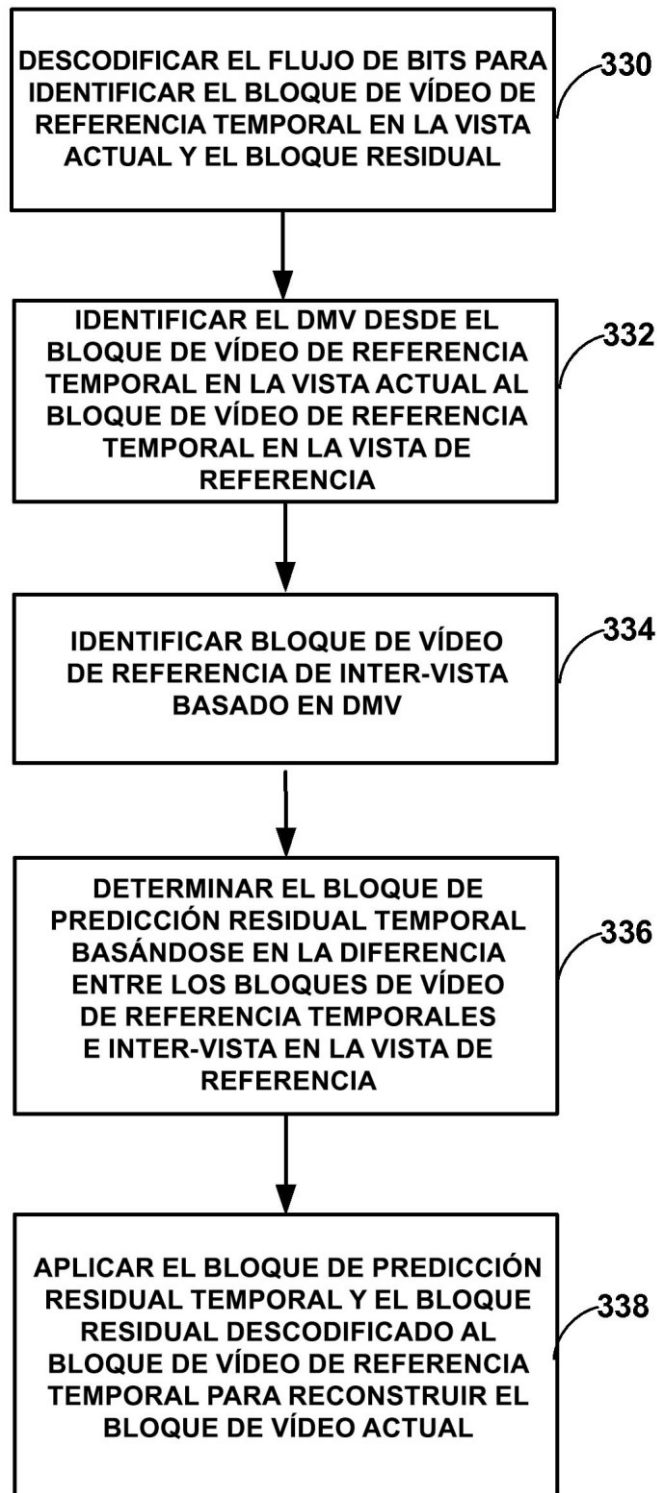


FIG. 18

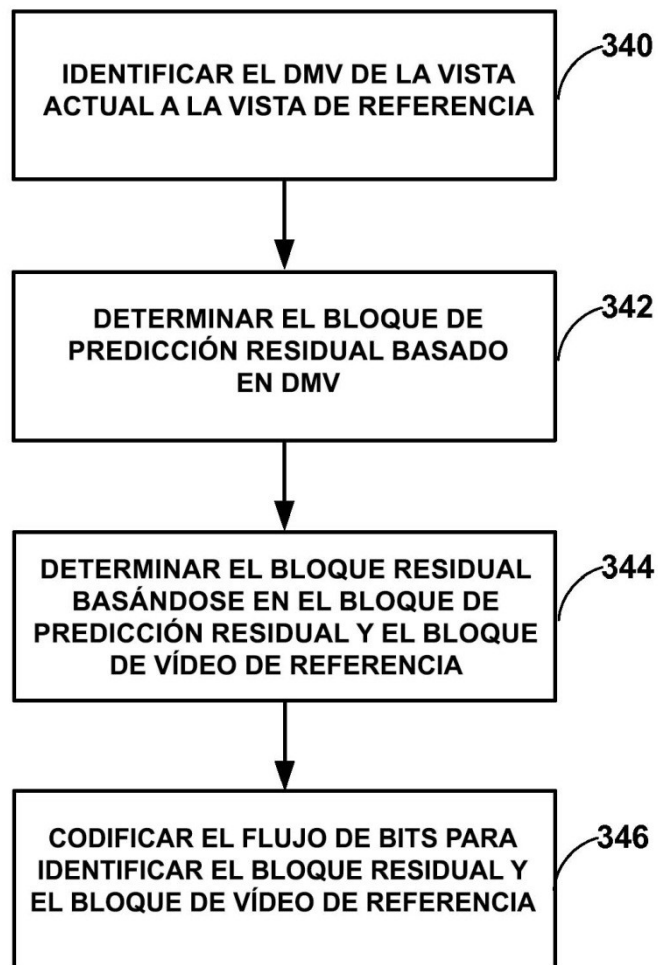


FIG. 19

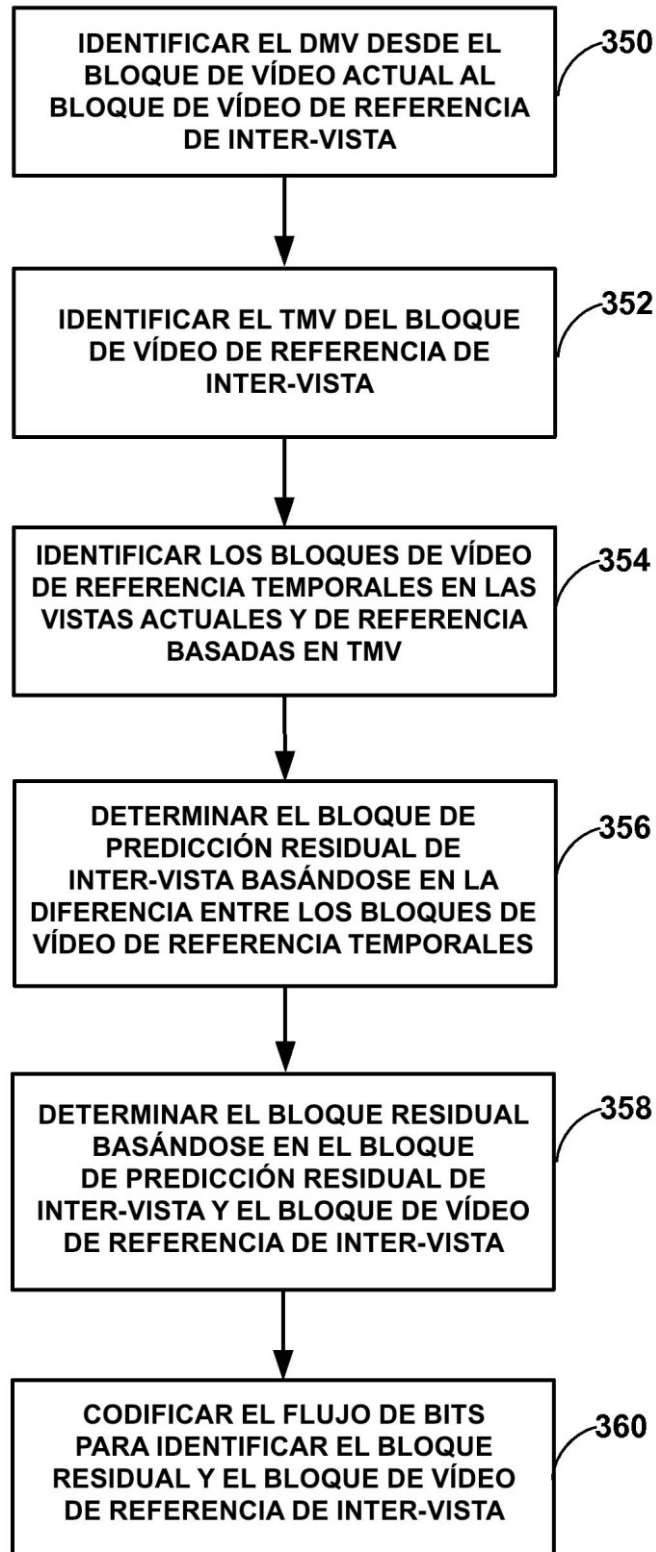


FIG. 20

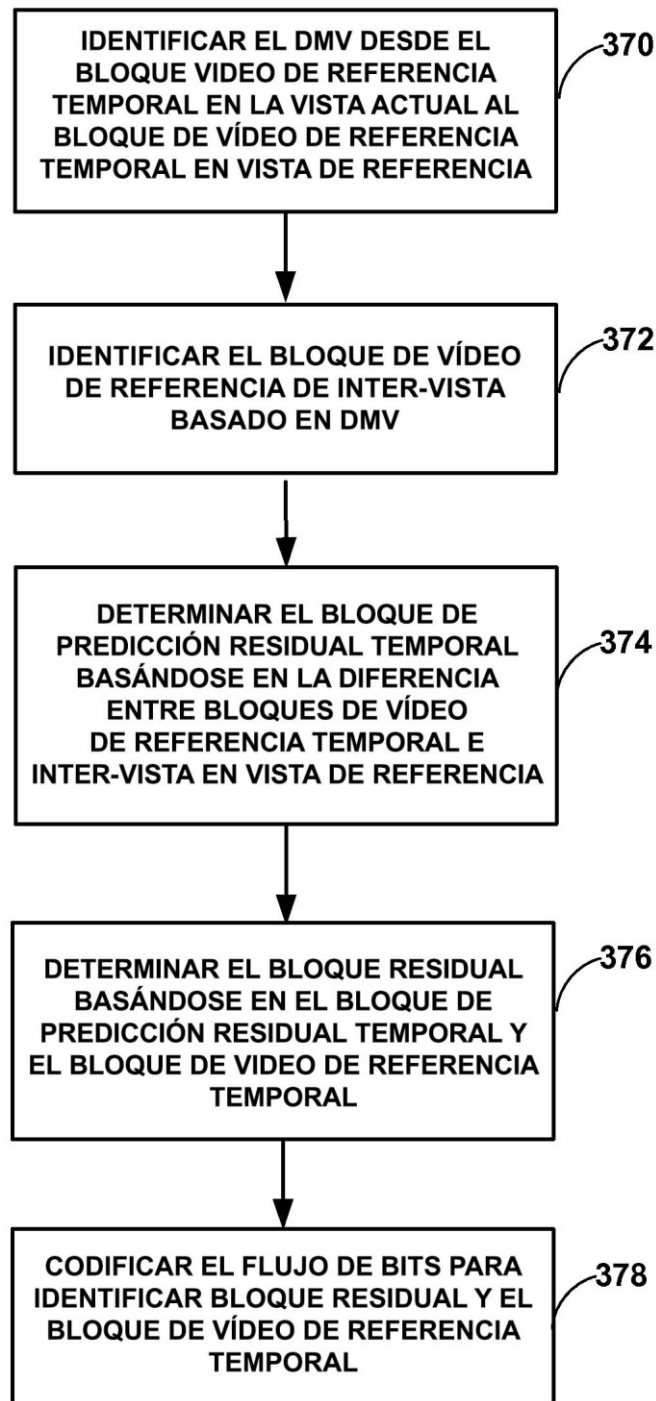


FIG. 21

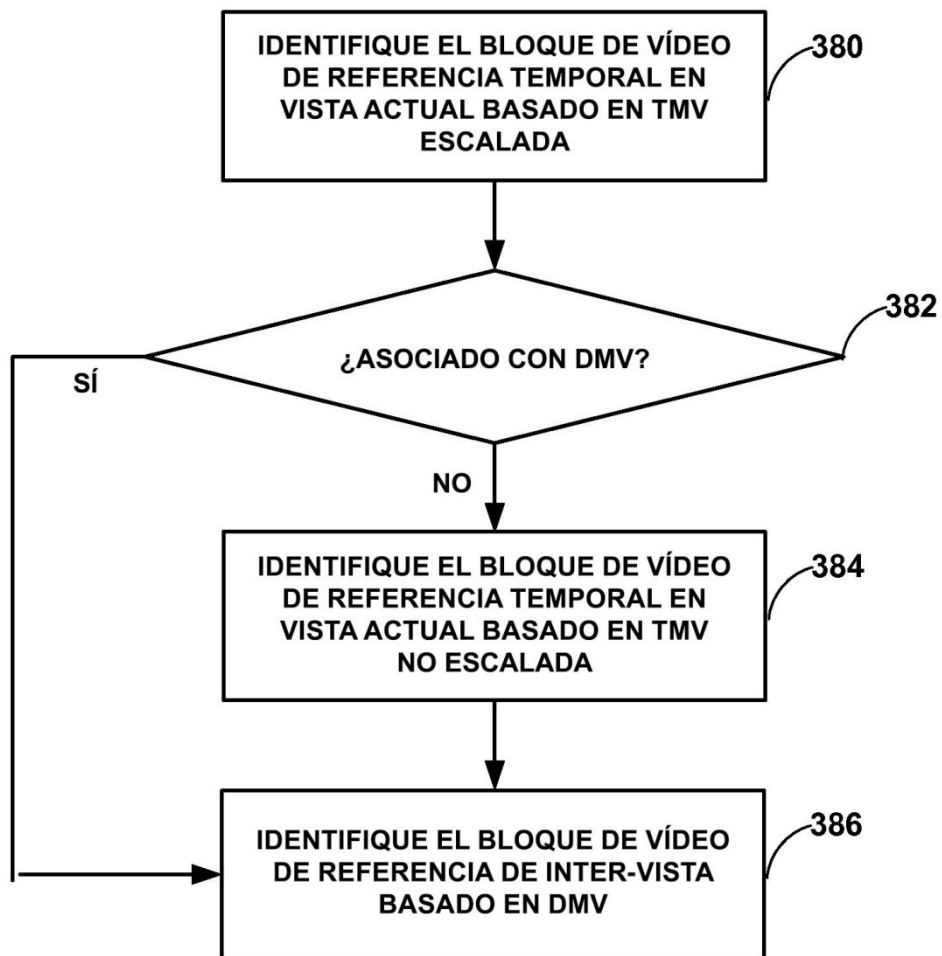


FIG. 22

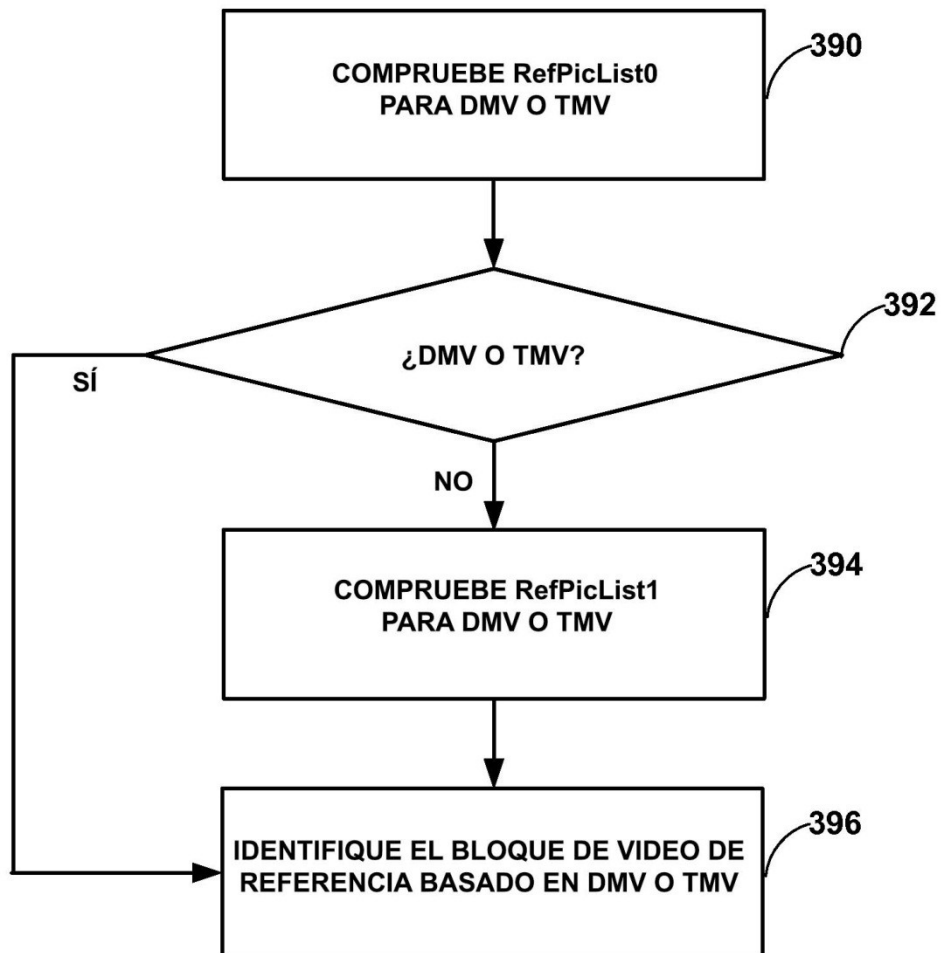


FIG. 23