



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS  
ESPAÑA



⑪ Número de publicación: **2 841 986**

⑮ Int. Cl.:

**H04N 19/51** (2014.01)

**H04N 19/597** (2014.01)

⑫

## TRADUCCIÓN DE PATENTE EUROPEA

T3

⑥ Fecha de presentación y número de la solicitud internacional: **17.07.2014 PCT/US2014/047000**

⑦ Fecha y número de publicación internacional: **22.01.2015 WO15009914**

⑨ Fecha de presentación y número de la solicitud europea: **17.07.2014 E 14748047 (9)**

⑩ Fecha y número de publicación de la concesión europea: **07.10.2020 EP 3022928**

---

④ Título: **Identificación de bloques usando vector de disparidad en la codificación de vídeo**

⑩ Prioridad:

**17.07.2013 US 201361847522 P  
16.07.2014 US 201414333084**

⑤ Fecha de publicación y mención en BOPI de la traducción de la patente:

**12.07.2021**

⑦ Titular/es:

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

⑧ Inventor/es:

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

⑨ Agente/Representante:

**FORTEA LAGUNA, Juan José**

**ES 2 841 986 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

Identificación de bloques usando vector de disparidad en la codificación de vídeo

## 5 CAMPO TÉCNICO

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

## 10 ANTECEDENTES

[0002] Las capacidades de vídeo digital se pueden incorporar a una amplia gama de dispositivos, incluyendo televisiones digitales, sistemas de radiodifusión directa digital, sistemas de radiodifusión 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 móviles o de radio por satélite, los denominados "teléfonos inteligentes", dispositivos de videoconferencia, dispositivos de transmisión continua de vídeo y similares. Los dispositivos de vídeo digital implementan técnicas de codificación de vídeo, tales como las descritas en las normas definidas por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, parte 10, Codificación avanzada de vídeo (AVC), la norma de Codificación de vídeo de alta eficacia (HEVC), actualmente en elaboración, y las ampliaciones de dichas normas. Los dispositivos de vídeo pueden transmitir, recibir, codificar, descodificar y/o almacenar información de vídeo digital más eficazmente, implementando dichas técnicas de codificación de vídeo.

[0003] Las técnicas de codificación de vídeo incluyen la predicción espacial (intraimagen) y/o la predicción temporal (interimagen) para reducir o eliminar la redundancia intrínseca en las secuencias de vídeo. Para la codificación de vídeo basada en bloques, un fragmento de vídeo (por ejemplo, una trama de vídeo o una porción de una trama de vídeo) se puede dividir en bloques de vídeo, que también se pueden denominar bloques arbolados, unidades de codificación (CU) y/o nodos de codificación. Los bloques de vídeo en un fragmento intracodificado (I) de una imagen se codifican usando predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen. Los bloques de vídeo en un fragmento intercodificado (P o B) de una imagen pueden usar predicción espacial con respecto a muestras de referencia de bloques vecinos en la misma imagen o predicción temporal con respecto a muestras de referencia en otras imágenes de referencia. Las imágenes se pueden denominar tramas y las imágenes de referencia se pueden denominar tramas de referencia.

[0004] La predicción espacial o temporal da como resultado un bloque predictivo para un bloque que se va a codificar. Los datos residuales representan diferencias de píxeles entre el bloque original que se va a codificar y el bloque predictivo. Un bloque intercodificado se codifica de acuerdo con un vector de movimiento que apunta a un bloque de muestras de referencia que forman el bloque predictivo, y los datos residuales que indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intracodificado se codifica de acuerdo con un modo de intracodificación y los datos residuales. Para compresión adicional, los datos residuales se pueden transformar desde el dominio de píxel en un dominio de transformada, dando como resultado coeficientes de transformada residuales, que a continuación se pueden cuantificar. Los coeficientes de transformada cuantificados, dispuestos inicialmente en una matriz bidimensional, se pueden explorar para producir un vector unidimensional de coeficientes de transformada, y se puede aplicar la codificación por entropía para lograr aún más compresión. Las técnicas de codificación y descodificación de vídeo ejemplares son TECH G ET AL.: "3D-HEVC Test Model 4 [Modelo de prueba 3D-HEVC 4]", 4. CONFERENCIA JCT-3V; 20-4-2013 - 26-4-2013; INCHEON; (EL EQUIPO DE COLABORACIÓN CONJUNTA EN LA ELABORACIÓN DE LA AMPLIACIÓN DE CODIFICACIÓN DE VÍDEO 3D DE ISO/IEC JTC1/SC29/WG11 E ITU-T SG.16); (publicado el 17-06-2013) y el documento US2011/0044550 A1.

## 50 BREVE EXPLICACIÓN

[0005] La presente invención se divulga de acuerdo con las reivindicaciones adjuntas. La presente divulgación describe técnicas para identificar un bloque de referencia (también denominado bloque correspondiente) con un vector de disparidad de un bloque actual en un procedimiento de codificación de vídeo. El bloque de referencia se localiza en una vista diferente a la vista que se incluye en el bloque actual. En algunos ejemplos, las técnicas pueden definir una posición de inicio del vector de disparidad del bloque actual. Por ejemplo, las componentes x e y del vector de disparidad se pueden sumar a o restar de las coordenadas x e y de la posición de inicio del vector de disparidad para determinar una localización de un bloque de referencia al que se hace referencia por el vector de disparidad. La eficacia o las ganancias de codificación de vídeo se pueden obtener identificando el bloque de referencia con un vector de disparidad con una posición de inicio descrita en la presente divulgación.

[0006] Los detalles de uno o más ejemplos se exponen en los dibujos adjuntos y la descripción a continuación. Otros rasgos característicos, objetivos y ventajas resultarán evidentes a partir de la descripción, los dibujos y las reivindicaciones.

**BREVE DESCRIPCIÓN DE LOS DIBUJOS****[0007]**

5 La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y descodificación de vídeo de ejemplo que puede implementar o utilizar de otro modo técnicas para la identificación de bloques.

10 La FIG. 2 es un diagrama conceptual que ilustra la codificación de una secuencia de codificación de vídeo multivista.

15 La FIG. 3 es un diagrama conceptual que ilustra un patrón de predicción de codificación de vídeo multivista de ejemplo.

20 La FIG. 4 es un diagrama conceptual que ilustra posiciones de inicio de ejemplo para un vector de disparidad para identificar un bloque de referencia.

25 La FIG. 5 es un diagrama conceptual que ilustra bloques vecinos temporales para la derivación del vector de disparidad basado en bloques vecinos (NBDV).

30 La FIG. 6 es un diagrama conceptual que ilustra la derivación de bloques de profundidad a partir de una vista de referencia asociada con la predicción de síntesis de vista hacia atrás (BVSP).

35 La FIG. 7 es un diagrama de bloques que ilustra la derivación del candidato de vector de movimiento predicho entre vistas.

40 La FIG. 8 es un diagrama conceptual que ilustra la derivación de un candidato de vector de movimiento heredado (MVI) para codificación de profundidad.

45 La FIG. 9 es un diagrama conceptual que ilustra una estructura de predicción de ejemplo de predicción residual avanzada.

50 La FIG. 10 es otro diagrama conceptual que ilustra una estructura de predicción de ejemplo de predicción residual avanzada.

55 La FIG. 11 es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo que puede implementar o utilizar de otro modo técnicas para la identificación de bloques de acuerdo con una o más técnicas de ejemplo descritas en la presente divulgación.

60 La FIG. 12 es un diagrama de bloques que ilustra un ejemplo de un descodificador de vídeo que puede implementar o utilizar de otro modo técnicas para la identificación de bloques de acuerdo con una o más técnicas de ejemplo descritas en la presente divulgación.

65 La FIG. 13 es un diagrama de flujo que ilustra un procedimiento de ejemplo de descodificación de vídeo.

70 La FIG. 14 es un diagrama de flujo que ilustra un procedimiento de ejemplo de codificación de vídeo.

**45 DESCRIPCIÓN DETALLADA**

**[0008]** La presente divulgación describe diversas técnicas para codificar contenido de vídeo tridimensional (3D) que incluye vistas de textura y vistas de profundidad. Más específicamente, cuando se aplica la predicción de movimiento entre vistas para identificar un bloque en la referencia, se proporcionan diferentes procedimientos. Las técnicas se pueden realizar, en algunos aspectos, por un codificador de vídeo. En otros aspectos, las técnicas se pueden realizar por un descodificador de vídeo. Además, dichos procedimientos se podrían realizar en otros dispositivos, tales como transcodificadores, elementos de red con reconocimiento de medios (MANE) o similares. En la presente divulgación, las técnicas se describirán con respecto a los codificadores y descodificadores de vídeo para propósitos ilustrativos. Por ejemplo, los codificadores y descodificadores de vídeo pueden utilizar herramientas de codificación que requieren acceso a un bloque correspondiente en una vista de referencia asociada con el bloque actual, tal como en la codificación residual avanzada o la codificación de predicción entre vistas.

**[0009]** En algunos ejemplos de codificación de vídeo (por ejemplo, codificación o descodificación), un codificador de vídeo (por ejemplo, un codificador de vídeo o un descodificador de vídeo) determina un vector de disparidad para un bloque actual que se va a codificar (por ejemplo, codificar o descodificar). El vector de disparidad se refiere a un bloque de referencia (también denominado bloque correspondiente) en una vista distinta de la vista que incluye el bloque actual. El vector de disparidad se usa por diversas herramientas de codificación de vídeo que se usan para codificar o descodificar el bloque actual, tal como predicción de movimiento entre vistas, predicción residual entre vistas, y herramientas de codificación de vídeo donde el vector de disparidad se convierte en un vector de movimiento de disparidad, como unos pocos ejemplos. Las técnicas descritas en la presente divulgación

describen modos en los que identificar el bloque de referencia en base al vector de disparidad para el bloque actual.

5 [0010] Para identificar el bloque de referencia, el codificador de vídeo puede sumar las coordenadas de una posición de inicio a los valores del vector de disparidad. Por ejemplo, el vector de disparidad incluye una componente x y una componente y. El codificador de vídeo suma la coordenada x de la posición de inicio a la componente x del vector de disparidad, y suma la coordenada y de la posición de inicio a la coordenada y del vector de disparidad. Los valores resultantes pueden ser la coordenada x y la coordenada y de un píxel en una posición final. El codificador de vídeo puede determinar un bloque que engloba una localización (es decir, el píxel) 10 con los valores de coordenadas resultantes e identificar ese bloque como el bloque de referencia para el bloque actual.

15 [0011] En algunos ejemplos, el codificador de vídeo puede establecer la posición de inicio del vector de disparidad de modo que el vector de disparidad se refiera a un píxel localizado en la localización C0 en la imagen de referencia de la vista de referencia. El codificador de vídeo determina la localización C0 en la imagen de referencia de la vista de referencia en base al valor del vector de disparidad (por ejemplo, en base a la componente x y la componente y del vector de disparidad).

20 [0012] Una ecuación para determinar la localización C0, basada en el valor del vector de disparidad, en la imagen de referencia de la vista de referencia se describe con más detalle a continuación. En general, la posición de inicio del vector de disparidad se determina de modo que el vector de disparidad se refiere a la localización C0 como que es el píxel superior izquierdo en un subbloque central del bloque actual. El centro del bloque actual incluye cuatro píxeles que se pueden considerar como que forman un subbloque 2x2 (por ejemplo, el subbloque central). Si la posición de inicio del vector de disparidad es el píxel superior izquierdo en este subbloque 2x2, entonces el 25 vector de disparidad se refiere a la localización C0 en la imagen de referencia de la vista de referencia (por ejemplo, el punto final del vector de disparidad es la localización C0 en la imagen de referencia).

30 [0013] En algunos ejemplos, la localización del subbloque central 2x2 se puede calcular dividiendo la longitud y el ancho del bloque por dos. La coordenada x de la esquina superior izquierda del subbloque central 2x2 es la coordenada x de la esquina superior izquierda del bloque actual más el ancho dividido por dos menos uno. La coordenada y de la esquina superior izquierda del subbloque central 2x2 es la coordenada y de la esquina superior izquierda del bloque actual más la altura dividida por dos menos uno. La esquina superior derecha del subbloque central 2x2 es el píxel localizado a la derecha inmediata del píxel en la esquina superior izquierda del subbloque central 2x2, la esquina inferior izquierda del subbloque central 2x2 es el píxel localizado inmediatamente debajo 35 del píxel en la esquina superior izquierda del subbloque central 2x2, y la esquina inferior derecha del subbloque central 2x2 es el píxel localizado a la derecha inmediata del píxel en la esquina inferior izquierda, inmediatamente debajo del píxel en la esquina superior derecha, y en diagonal a la parte inferior derecha de la esquina superior izquierda del subbloque central 2x2.

40 [0014] El codificador de vídeo puede determinar un bloque en la imagen de referencia que cubre (es decir, incluye) el píxel localizado en la localización C0. A continuación, el codificador de vídeo puede identificar el bloque determinado como que es el bloque de referencia (también denominado bloque correspondiente) para el bloque actual que se usa para las herramientas de codificación de vídeo que se basan en un vector de disparidad (por ejemplo, para la predicción de movimiento entre vistas, la predicción residual entre vistas y/o la predicción de 45 síntesis de vistas basada en bloques, como unos pocos ejemplos no limitantes).

50 [0015] Sin embargo, la utilización del bloque que cubre el píxel localizado en la localización C0 como bloque de referencia puede no proporcionar eficacias de codificación suficientes. Por ejemplo, las tendencias estadísticas pueden indicar que un bloque que cubre un píxel localizado en una localización distinta a la localización C0 puede ser un mejor candidato como bloque de referencia para el bloque actual. En consecuencia, establecer una posición de inicio de un vector de disparidad igual a una localización de un píxel superior izquierdo del subbloque central 2x2 dentro del bloque actual puede dar como resultado la identificación de un bloque que puede no proporcionar 55 eficacias de codificación óptimas.

60 [0016] La presente divulgación describe técnicas de ejemplo para determinar una localización de un píxel distinto de un píxel localizado en la localización C0 en una imagen de referencia de la vista de referencia, e identificar un bloque que cubre ese otro píxel como un bloque de referencia (por ejemplo, el bloque correspondiente) para un bloque actual. Como un ejemplo, la divulgación describe técnicas para determinar una localización de un píxel en la imagen de referencia localizada en la parte inferior derecha del píxel localizado en la localización C0. Por ejemplo, el píxel localizado en la parte inferior derecha del píxel localizado en la localización C0 se puede considerar como que está localizado en (1, 1) en relación con, y no en términos absolutos dentro de la imagen de referencia, la localización del píxel localizado en C0.

65 [0017] La localización del píxel en la imagen de referencia localizada en la parte inferior derecha del píxel localizado en C0 se denomina localización C3 en la presente divulgación. El codificador de vídeo determina la localización C3 en base al valor del vector de disparidad (por ejemplo, las componentes x e y del vector de

disparidad), y una ecuación para determinar la localización C3 que se describe con más detalle a continuación. En los ejemplos donde el codificador de vídeo determina los píxeles localizados en la localización C3, el codificador de vídeo puede determinar el bloque que cubre (es decir, incluye) el píxel localizado en la localización C3 e identificar el bloque determinado como el bloque de referencia para el bloque actual.

- 5 [0018] Como se describe anteriormente, la localización C3 está localizada un píxel a la derecha y un píxel hacia la parte inferior de la localización C0. Por lo tanto, si la posición de inicio del vector de disparidad dentro del bloque actual también se cambia un píxel a la derecha y un píxel hacia la parte inferior, el vector de disparidad ahora se referirá a la localización C3. Por ejemplo, como se describe anteriormente, si el vector de disparidad comienza desde el píxel superior izquierdo del subbloque 2x2 central dentro del bloque actual, entonces el vector de disparidad se refiere a la localización C0 en la imagen de referencia de la vista de referencia. En consecuencia, si la posición de inicio del vector de disparidad se desplaza en un píxel a la derecha y un píxel a la parte inferior (es decir, el píxel inferior derecho del subbloque 2x2 central dentro del bloque actual), entonces el vector de disparidad se refiere a la localización C3.
- 10 [0019] En una o más técnicas de ejemplo descritas en la presente divulgación, el codificador de vídeo puede determinar el bloque dentro de la imagen de referencia de la vista de referencia que cubre la localización a la que se hace referencia por el vector de disparidad comenzando desde el píxel inferior derecho del subbloque 2x2 central, e identificar ese bloque como el bloque de referencia, en lugar de determinar el bloque que cubre la localización a la que se hace referencia por el vector de disparidad que comienza desde el píxel superior izquierdo del subbloque 2x2 central. En otras palabras, el codificador de vídeo puede identificar el bloque en la imagen de referencia en la vista de referencia que cubre la localización C3, a la que se hace referencia por un vector de disparidad que comienza desde el píxel inferior derecho del subbloque 2x2 central, como el bloque de referencia, en lugar del bloque que cubre la localización C0, al que se hace referencia por un vector de disparidad que comienza desde el píxel superior izquierdo del subbloque 2x2 central.
- 15 [0020] En algunos ejemplos, en lugar de determinar la localización C3, el codificador de vídeo puede determinar la localización del píxel localizado a la derecha del píxel localizado en la localización C0 (denominado píxel localizado en la localización C1) comenzando el vector de disparidad desde el píxel superior derecho del subbloque 2x2 central o determinar la localización del píxel localizado debajo del píxel localizado en la localización C0 (denominado píxel localizado en la localización C2) comenzando el vector de disparidad desde el píxel inferior izquierdo del subbloque 2x2 central. En estos ejemplos, el codificador de vídeo puede determinar el bloque en la imagen de referencia en la vista de referencia que cubre el píxel localizado en la localización C1 o el píxel localizado en la localización C2, e identificar el bloque determinado como el bloque de referencia para el bloque actual.
- 20 [0021] En los ejemplos anteriores, para determinar el píxel localizado en una de las localizaciones C0, C1, C2 o C3, el codificador de vídeo puede establecer la posición de inicio del vector de disparidad de modo que el vector de disparidad apunte al píxel localizado en una de las localizaciones C0, C1, C2 o C3. Por ejemplo, el codificador de vídeo suma los valores de coordenada de la posición de inicio al valor del vector de disparidad de modo que los valores de coordenada resultantes sean los de una de las localizaciones C0, C1, C2 o C3. De este modo, al establecer apropiadamente la posición de inicio del vector de disparidad, el codificador de vídeo puede hacer que el vector de disparidad haga referencia a uno de los píxeles localizados en una de las localizaciones C0, C1, C2 o C3.
- 25 [0022] Si el bloque que cubre el píxel localizado en la localización C3 en la imagen de referencia de la vista de referencia no incluye información de movimiento (por ejemplo, se intrapredice) o la información de movimiento no está disponible para la herramienta de codificación de vídeo que se basa en el vector de disparidad y que se usa para codificar el bloque actual, el codificador de vídeo puede determinar que esta herramienta de codificación de vídeo no está disponible para el bloque actual. Sin embargo, las técnicas no están limitadas así.
- 30 [0023] En algunos ejemplos, el codificador de vídeo puede determinar el bloque en la imagen de referencia de la vista de referencia que cubre el píxel localizado en la localización C3, y si la información de movimiento para el bloque asociado con la localización C3 no está disponible para una o más de las herramientas de codificación de vídeo que se basan en un vector de disparidad (por ejemplo, porque el bloque se intrapredice o la información de movimiento no es utilizable), el codificador de vídeo puede determinar posteriormente el bloque en la imagen de referencia de la vista de referencia que cubre el píxel localizado en la localización C0. A continuación, el codificador de vídeo puede identificar el bloque que cubre el píxel localizado en la localización C0 como bloque de referencia. En algunos ejemplos, el codificador de vídeo puede realizar las etapas opuestas (por ejemplo, si el bloque que cubre el píxel localizado en la localización C0 no incluye información de movimiento utilizable para herramientas de codificación de vídeo que se basan en el vector de disparidad, el codificador de vídeo identifica el bloque que cubre el píxel localizado en la localización C3 como bloque de referencia).
- 35 [0024] Si el bloque que cubre el píxel localizado en la localización C3 y el bloque que cubre el píxel localizado en la localización C0 no incluyen información de movimiento que está disponible para las herramientas de codificación de vídeo que se basan en el vector de disparidad, el codificador de vídeo puede determinar que una o más de las herramientas de codificación de vídeo que se basan en el vector de disparidad no están disponibles

para el bloque actual. En algunos ejemplos, en lugar de determinar que una o más de las herramientas de codificación de vídeo no están disponibles, el codificador de vídeo puede determinar si el bloque respectivo que cubre el píxel localizado en la localización C1 o la localización C2 incluye información de movimiento disponible para una o más de las herramientas de codificación de vídeo, e identificar un bloque de este tipo como el bloque de referencia para el bloque actual.

**[0025]** También puede haber opciones adicionales, tales como bloques localizados en las localizaciones BR0, BR1 o BR2 que se describen con más detalle a continuación. Por ejemplo, el codificador de vídeo puede determinar si la información de movimiento para un bloque localizado en las localizaciones BR0, BR1 o BR2 en la imagen de referencia de la vista de referencia incluye información de movimiento utilizable o deseable para una o más herramientas de codificación de vídeo que se basan en el vector de disparidad, e identificar el determinado de los bloques BR0, BR1 o BR2 como el bloque de referencia para el bloque actual.

**[0026]** La localización del bloque BR0 en la imagen de referencia de la vista de referencia es igual a la localización del bloque en la parte inferior derecha del bloque al que se hace referencia por el vector de disparidad si el vector de disparidad comienza desde la esquina superior izquierda del bloque actual. La localización del bloque BR1 está hacia la parte inferior derecha del bloque BR0, y la localización del bloque BR2 está hacia la parte inferior derecha del bloque BR1. Por ejemplo, en términos relativos, y no en una localización absoluta dentro de la imagen de referencia, el bloque BR1 está localizado en (1, 1) en relación con el bloque BR0, y el bloque BR2 está localizado en (2, 2) en relación con el bloque BR0.

**[0027]** En algunos ejemplos, el codificador de vídeo puede determinar una localización en la imagen de referencia de una vista de referencia en base al vector de disparidad comenzando desde la esquina superior izquierda del bloque actual. El codificador de vídeo se puede desplazar y cuantificar la localización determinada de modo que la localización determinada se alinee como la esquina superior izquierda de un bloque 4x4 en la imagen de referencia de la vista de referencia. A continuación, el codificador de vídeo puede desplazar además (por ejemplo, hacia la derecha, hacia abajo o una combinación de ambos) la localización determinada para determinar una localización de otro bloque dentro de la imagen de referencia. El codificador de vídeo puede identificar este bloque como el bloque de referencia para la imagen actual.

**[0028]** Las técnicas de ejemplo anteriores para identificar el bloque de referencia se describen con respecto al codificador de vídeo que determina el bloque que cubre el píxel en la localización C3, a continuación C0, a continuación C1, a continuación C2, a continuación el bloque en la localización BR0, a continuación BR1 y a continuación BR2, y a continuación las localizaciones desplazadas. Sin embargo, las técnicas descritas en la presente divulgación no están limitadas así. Son posibles otras permutaciones del orden en el que se comprueban los bloques en la imagen de referencia de la vista de referencia y se contemplan por las técnicas descritas en la presente divulgación.

**[0029]** Además, no es necesario que el codificador de vídeo compruebe cada uno de estos bloques para identificar el bloque de referencia. El codificador de vídeo puede interrumpir la búsqueda de un bloque de referencia tan pronto como el codificador de vídeo determine (es decir, en base a la determinación de la codificación de vídeo) que uno de estos bloques es adecuado para una o más de las herramientas de codificación de vídeo que se basan en el vector de disparidad.

**[0030]** En algunos ejemplos, el codificador de vídeo puede interrumpir la búsqueda de un bloque de referencia después de comprobar un subconjunto de estas localizaciones de ejemplo, incluyendo la comprobación solo de un bloque, incluso si no se identifica ningún bloque de referencia. Por ejemplo, el codificador de vídeo puede determinar si el bloque que cubre el píxel localizado en la localización C3 incluye información de movimiento disponible para una o más herramientas de codificación de vídeo. Si la información de movimiento no está disponible (por ejemplo, porque el bloque está intracodificado o el vector de movimiento del bloque es un vector de movimiento de disparidad o el vector de movimiento del bloque se refiere a una imagen que no está en la lista de imágenes de referencia del bloque actual), el codificador de vídeo puede interrumpir la búsqueda de un bloque de referencia y determinar que no hay ningún bloque de referencia disponible aunque pueda haber otros bloques posibles para comprobar.

**[0031]** Las técnicas anteriores para identificar el bloque de referencia al que se hace referencia por el vector de disparidad se describen por separado simplemente para ayudar a la comprensión. Sin embargo, se debe entender que las técnicas descritas en la presente divulgación no están limitadas así, y se pueden combinar conjuntamente. Además, las técnicas de ejemplo anteriores se pueden generalizar a cualquier supuesto cuando sea necesario identificar un elemento de sintaxis o una variable asociada con un bloque.

**[0032]** Además, las técnicas anteriores se describen con respecto a una técnica implícita para identificar el bloque de referencia (por ejemplo, el bloque de referencia es siempre el bloque que cubre el píxel localizado en la localización C3 o comprobar en primer lugar el bloque de referencia que cubre la localización C3, y a continuación moverse a la localización C0, etc.). Estas técnicas están implícitas en el sentido de que el codificador de vídeo no necesita señalizar información al descodificador de vídeo sobre cómo identificar el bloque de referencia (por

ejemplo, no es necesario señalización explícita de información sobre cómo identificar el bloque de referencia). En algunos ejemplos, el codificador de vídeo puede comprobar múltiples bloques que incluye cada uno información de movimiento para herramientas de codificación de vídeo que se basan en el vector de disparidad, y determinar qué bloque proporciona la mayor eficacia de codificación entre los múltiples bloques que se comprueban. A continuación, el codificador de vídeo puede señalizar explícitamente al descodificador de vídeo información indicativa del bloque que se usa como bloque de referencia a partir de la que el descodificador de vídeo puede determinar qué bloque debería ser el bloque de referencia.

5 [0033] Además, las técnicas anteriores se describen con respecto a las herramientas de codificación de vídeo que se basan en un vector de disparidad. Sin embargo, las técnicas descritas en la presente divulgación no están limitadas así. Por ejemplo, las técnicas anteriores se pueden generalizar a cualquier supuesto cuando sea necesario identificar un elemento de sintaxis o una variable asociada con un bloque.

10 [0034] En general, en las técnicas descritas en la presente divulgación, el codificador de vídeo puede determinar un vector de disparidad para un bloque actual en una imagen actual en una vista actual. El codificador de vídeo puede determinar un bloque en una imagen de referencia en una vista de referencia en base a una localización en la imagen de referencia a la que se refiere el vector de disparidad (por ejemplo, determinar el bloque que cubre la localización C3 en la imagen de referencia).

15 [0035] El codificador de vídeo puede codificar (por ejemplo, codificar o descodificar) por interpredicción el bloque actual en base al bloque determinado. Por ejemplo, como se describe con más detalle a continuación, en la predicción de movimiento entre vistas, el codificador de vídeo forma una lista de predictores de vector de movimiento. La información de movimiento del bloque determinado en la imagen de referencia puede ser uno de los predictores de vector de movimiento. Si la información del vector de movimiento del bloque determinado se selecciona para codificar por interpredicción (por ejemplo, codificar por interpredicción o descodificar por interpredicción) el bloque actual, entonces el codificador de vídeo codifica por interpredicción el bloque actual en base a la información de movimiento del bloque determinado en la imagen de referencia.

20 [0036] En algunos ejemplos, el codificador de vídeo puede convertir el vector de disparidad en un vector de movimiento de disparidad para el bloque actual. En estos ejemplos, el vector de movimiento de disparidad para el bloque actual se refiere al bloque de referencia, que se puede considerar como un bloque predictivo. Por ejemplo, el codificador de vídeo incluye una ruta de realimentación que almacena una reconstrucción del bloque de referencia para propósitos de codificación por interpredicción, y el codificador de vídeo almacenó una reconstrucción del bloque de referencia para propósitos de descodificación por interpredicción. La reconstrucción del bloque de referencia se puede considerar un bloque predictivo. El codificador de vídeo determina un residuo entre una reconstrucción de un bloque de referencia (por ejemplo, un bloque predictivo) y el bloque actual para codificar el bloque actual.

25 [0037] La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y descodificación de vídeo de ejemplo que se puede configurar para implementar o utilizar de otro modo técnicas para la identificación de bloques. Por ejemplo, dichas técnicas de identificación de bloques pueden ser útiles durante la predicción de movimiento entre vistas, la predicción residual entre vistas, ejemplos donde el vector de disparidad se convierte en un vector de movimiento de disparidad u otras herramientas de codificación de vídeo que se basan en un vector de disparidad. La predicción de movimiento entre vistas, la predicción residual entre vistas y los ejemplos donde el vector de disparidad se convierte en un vector de movimiento de disparidad se describen con más detalle a continuación.

30 [0038] Como se muestra en la FIG. 1, el sistema 10 incluye un dispositivo de origen 12 que proporciona datos de vídeo codificados que se van a descodificar en un momento posterior por un dispositivo de destino 14. En particular, el dispositivo de origen 12 proporciona los datos de vídeo al dispositivo de destino 14 por medio de un medio legible por ordenador 16. El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender cualquiera de una amplia gama de dispositivos, incluyendo ordenadores de escritorio, ordenadores plegables (es decir, portátiles), ordenadores de tableta, descodificadores, equipos telefónicos tales como los denominados teléfonos "inteligentes", los denominados paneles "inteligentes", televisiones, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de transmisión continua de vídeo o similares. En algunos casos, el dispositivo de origen 12 y el dispositivo de destino 14 pueden estar equipados para la comunicación inalámbrica.

35 [0039] El dispositivo de destino 14 puede recibir los datos de vídeo codificados que se van a descodificar por medio del enlace 16. El enlace 16 puede comprender cualquier tipo de medio o dispositivo que pueda mover los datos de vídeo codificados desde el dispositivo de origen 12 al dispositivo de destino 14. En un ejemplo, el enlace 16 puede comprender un medio de comunicación (medio por cable o inalámbrico) para posibilitar que el dispositivo de origen 12 transmita datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real. Los datos de vídeo codificados se pueden modular de acuerdo con una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitir al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrica o por cable, tal como un espectro de radiofrecuencia

(RF) o una o más líneas físicas de transmisión. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área amplia o una red global, tal como Internet. El medio de comunicación puede incluir encaminadores, comutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo de origen 12 al dispositivo de destino 14.

5 [0040] En algunos ejemplos, los datos codificados se pueden emitir desde la interfaz de salida 22 del dispositivo de origen 12 a un dispositivo de almacenamiento, tal como el dispositivo de almacenamiento 31. De forma similar, se puede acceder a los datos codificados desde el dispositivo de almacenamiento 31 por la interfaz de entrada 28 del dispositivo de destino 14. El dispositivo de almacenamiento 31 puede incluir cualquiera de una variedad de

10 medios de almacenamiento de datos de acceso local o distribuidos, tales como una unidad de disco duro, discos Blu-ray, DVD, CD-ROM, memoria flash, memoria volátil o no volátil o cualquier otro medio de almacenamiento digital adecuado para almacenar datos de vídeo codificados. En otro ejemplo, el dispositivo de almacenamiento 31

15 puede corresponder a un servidor de archivos o a otro dispositivo de almacenamiento intermedio que puede almacenar el vídeo codificado generado por el dispositivo de origen 12. El dispositivo de destino 14 puede acceder a datos de vídeo almacenados desde el dispositivo de almacenamiento por medio de transmisión continua o

descarga. El servidor de archivos puede ser cualquier tipo de servidor que pueda almacenar datos de vídeo codificados y transmitir esos datos de vídeo codificados al dispositivo de destino 14. Los servidores de archivos de ejemplo incluyen un servidor web (por ejemplo, para un sitio web), un servidor FTP, dispositivos de almacenamiento conectados en red (NAS) o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de

20 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 Wi-Fi), una conexión por cable (por ejemplo, DSL, módem de cable, etc.) o una combinación de ambas que sea adecuada para acceder a datos de vídeo codificados almacenados en un servidor de archivos. La transmisión de datos de vídeo codificados desde el dispositivo de

25 almacenamiento puede ser una transmisión continua, una transmisión de descarga o una combinación de ambas.

[0041] Las técnicas de la presente divulgación no están limitadas necesariamente a aplicaciones o configuraciones inalámbricas. Las técnicas se pueden aplicar a la codificación de vídeo como apoyo de cualquiera de una variedad de aplicaciones de multimedia, tales como radiodifusiones de televisión por el aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo por transmisión continua

30 por internet, tales como la transmisión continua adaptativa dinámica por HTTP (DASH), vídeo digital que se codifica en un medio de almacenamiento de datos, descodificación de vídeo digital almacenado en un medio de almacenamiento de datos u otras aplicaciones. En algunos ejemplos, el sistema 10 se puede configurar para

35 admitir una transmisión de vídeo unidireccional o bidireccional para admitir aplicaciones tales como transmisión continua de vídeo, reproducción de vídeo, radiodifusión de vídeo y/o videotelefonía.

[0042] En el ejemplo de la FIG. 1, el dispositivo de origen 12 incluye una fuente de vídeo 18, un codificador de

vídeo 20 y una interfaz de salida 22. El dispositivo de destino 14 incluye una interfaz de entrada 28, un descodificador de vídeo 30 y un dispositivo de visualización 32. De acuerdo con la presente divulgación, el codificador de vídeo 20 del dispositivo de origen 12 y el descodificador de vídeo 30 del dispositivo de destino 14

40 se pueden configurar para implementar las técnicas de ejemplo descritas en la presente divulgación. En algunos ejemplos, un dispositivo de origen y un dispositivo de destino pueden incluir otros componentes o disposiciones. Por ejemplo, el dispositivo de origen 12 puede recibir datos de vídeo desde una fuente de vídeo 18 externa, tal

45 como una cámara externa. Del mismo modo, el dispositivo de destino 14 puede interactuar con un dispositivo de visualización externo, en lugar de incluir un dispositivo de visualización integrado.

[0043] El sistema 10 ilustrado de la FIG. 1 es simplemente un ejemplo, y las técnicas descritas en la presente divulgación se pueden realizar por cualquier dispositivo de codificación y/o descodificación de vídeo digital. Aunque, en general, las técnicas de la presente divulgación se realizan por un dispositivo de

50 codificación/descodificación de vídeo, las técnicas también se pueden realizar por un codificador/descodificador de vídeo, denominado típicamente "CÓDEC". Además, las técnicas de la presente divulgación también se pueden

realizar por un preprocesador de vídeo. El dispositivo de origen 12 y el dispositivo de destino 14 son simplemente ejemplos de dichos dispositivos de codificación en los que el dispositivo de origen 12 genera datos de vídeo

55 codificados para su transmisión al dispositivo de destino 14. En algunos ejemplos, los dispositivos 12, 14 pueden funcionar de manera sustancialmente simétrica, de modo que cada uno de los dispositivos 12, 14 incluye componentes de codificación y descodificación de vídeo. Por consiguiente, el sistema 10 puede admitir transmisión

de vídeo unidireccional o bidireccional entre los dispositivos de vídeo 12, 14, (por ejemplo para transmisión continua

de vídeo, reproducción de vídeo, radiodifusión de vídeo o videotelefonía).

[0044] La fuente de vídeo 18 del dispositivo de origen 12 puede incluir un dispositivo de captura de vídeo, tal

60 como una cámara de vídeo, un archivo de vídeo que contiene vídeo capturado previamente y/o una interfaz de suministro de vídeo para recibir vídeo desde un proveedor de contenidos de vídeo. Como otra alternativa, la fuente de vídeo 18 puede generar datos basados en gráficos de ordenador como vídeo de origen, o una combinación de

65 vídeo en directo, vídeo archivado y vídeo generado por ordenador. En algunos casos, si la fuente de vídeo 18 es una cámara de vídeo, el dispositivo de origen 12 y el dispositivo de destino 14 pueden formar los denominados

teléfonos con cámara o videoteléfonos. Sin embargo, como se menciona anteriormente, las técnicas descritas en

la presente divulgación pueden ser aplicables a la codificación de vídeo en general, y se pueden aplicar a

aplicaciones inalámbricas y/o por cable. En cada caso, se puede codificar el vídeo capturado, precapturado o generado por ordenador por el codificador de vídeo 20. A continuación, se puede emitir la información de vídeo codificada por la interfaz de salida 22 a un enlace 16.

5      **[0045]** La interfaz de entrada 28 del dispositivo de destino 14 recibe información del enlace 16 y/o del dispositivo de almacenamiento 31. La información recibida puede incluir información de sintaxis definida por el codificador de vídeo 20, que también se usa por el descodificador de vídeo 30, que incluye elementos de sintaxis que describen características y/o procesamiento de bloques y otras unidades codificadas. El dispositivo de visualización 32 muestra los datos de vídeo descodificados a un usuario, y puede comprender cualquiera de una variedad de dispositivos de visualización, tales como un tubo de rayos catódicos (CRT), una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de visualización.

10     **[0046]** El codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con una norma de codificación de vídeo, tal como la norma de Codificación de vídeo de alta eficacia (HEVC), y se pueden ajustar, en general, al modelo de prueba de HEVC (HM). La norma HEVC está elaborada por el Equipo de colaboración conjunta en codificación de vídeo (JCT-VC) del Grupo de expertos en codificación de vídeo (VCEG) de la ITU-T y el Grupo de expertos en imágenes en movimiento (MPEG) de la ISO/IEC. La especificación en texto más reciente de HEVC, borrador 10 (WD10), está disponible en [http://phenix.int-evry.fr/jct\\_end\\_user/documents/12\\_Geneva/wg11/JCTVC-L1003-v34.zip](http://phenix.int-evry.fr/jct_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v34.zip).

15     **[0047]** En algunos ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con diversas ampliaciones actualmente en elaboración tales como MV-HEVC, 3D-HEVC, 3D-AVC u otras normas para la codificación de multivista. De forma alternativa, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con otras normas privadas o industriales, tales como la norma ITU-T H.264, denominada de forma alternativa MPEG-4, parte 10, Codificación avanzada de vídeo (AVC) o ampliaciones de dichas normas. Sin embargo, las técnicas de la presente divulgación no se limitan a ninguna norma de codificación particular.

20     **[0048]** Otros ejemplos de normas de codificación de vídeo incluyen MPEG-2, ITU-T H.263, 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 escalable (SVC) y codificación de vídeo multivista (MVC). El codificador de vídeo 20 y el descodificador de vídeo 30 se pueden configurar de acuerdo con una o más de las normas de ejemplo anteriores, o pueden no configurarse de acuerdo con ninguna norma particular.

25     **[0049]** Aunque no se muestra en la FIG. 1, en algunos aspectos, el codificador de vídeo 20 y el descodificador de vídeo 30 se pueden integrar cada uno con un codificador y un descodificador de audio, y pueden incluir unidades MUX-DEMUX apropiadas, u otro hardware y software, para manejar la codificación tanto de audio como de vídeo en un flujo de datos común o en flujos de datos separados. Si fuera aplicable, las unidades MUX-DEMUX se pueden ajustar al protocolo multiplexor ITU H.223 u otros protocolos tales como el protocolo de datagramas de usuario (UDP).

30     **[0050]** La norma ITU-T H.264/MPEG-4 (AVC) se formuló por el Grupo de expertos en codificación de vídeo (VCEG) de la ITU-T conjuntamente con el Grupo de expertos en imágenes en movimiento (MPEG) de la ISO/IEC, como el producto de una asociación colectiva conocida como el Equipo mixto de vídeo (JVT). En algunos aspectos, las técnicas descritas en la presente divulgación se pueden aplicar a dispositivos que se ajustan en general a la norma H.264. La norma H.264 está descrita en la recomendación H.264 de la ITU-T, codificación avanzada de vídeo para servicios audiovisuales genéricos, por el grupo de estudio de la ITU-T, y con fecha de marzo de 2005, que se puede denominar en el presente documento norma H.264 o especificación H.264, o norma o especificación H.264/AVC. El Equipo mixto de vídeo (JVT) continúa trabajando en ampliaciones para H.264/MPEG-4 AVC.

35     **[0051]** El codificador de vídeo 20 y el descodificador de vídeo 30 se pueden implementar cada uno como cualquiera de una variedad de circuitos de codificador adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), matrices de puertas programables *in situ* (FPGA), lógica discreta, circuito integrado (IC), software, hardware, firmware o cualquier combinación de los mismos. Cada uno del codificador de vídeo 20 y el descodificador de vídeo 30 se puede incluir en uno o más codificadores o descodificadores, de los que cualquiera se puede integrar como parte de un codificador/descodificador (CÓDEC) de vídeo combinado. Un dispositivo que incluye el codificador de vídeo 20 y/o el descodificador de vídeo 30 puede comprender un circuito integrado, un microprocesador y/o un dispositivo de comunicación inalámbrica, tal como un teléfono móvil.

40     **[0052]** Cuando las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio de almacenamiento legible por ordenador adecuado y ejecutar las instrucciones en hardware usando uno o más procesadores para realizar las técnicas de la presente divulgación. Cada uno del codificador de vídeo 20 y el descodificador de vídeo 30 se puede incluir en uno o más codificadores

o descodificadores, de los que cualquiera se puede integrar como parte de un codificador/descodificador (CÓDEC) combinado en un dispositivo respectivo.

5 **[0053]** Los trabajos de normalización de HEVC se basan en un modelo en evolución de un dispositivo de codificación de vídeo denominado modelo de prueba de HEVC (HM). El HM supone varias capacidades adicionales de los dispositivos de codificación de vídeo en relación con los dispositivos existentes de acuerdo con, por ejemplo, ITU-T H.264/AVC. Por ejemplo, mientras que H.264 proporciona nueve modos de codificación por intrapredicción, el HM puede proporcionar hasta treinta y tres modos de codificación por intrapredicción.

10 **[0054]** En general, el HM describe que una trama o imagen de vídeo se puede dividir en una secuencia de bloques arbolados o máximas unidades de codificación (LCU) que incluyen tanto muestras de luma como de croma. La norma HEVC también se refiere a las LCU como "unidades de árbol de codificación" o CTU. Los datos de sintaxis dentro de un flujo de bits pueden definir un tamaño para la LCU, que es la máxima unidad de codificación en términos del número de píxeles. Un fragmento incluye un número de bloques arbolados consecutivos en orden de codificación. Una trama o imagen de vídeo se puede dividir en uno o más fragmentos. Cada bloque arbolado se puede separar en unidades de codificación (CU) de acuerdo con un árbol cuaternario. En general, una estructura de datos de árbol cuaternario incluye un nodo por CU, con un nodo raíz correspondiente al bloque arbolado. Si una CU se separa en cuatro sub-CU, el nodo correspondiente a la CU incluye cuatro nodos hoja, de los que cada uno corresponde a una de las sub-CU.

15 **[0055]** Cada nodo de la estructura de datos de árbol cuaternario puede proporcionar datos de sintaxis para la CU correspondiente. Por ejemplo, un nodo en el árbol cuaternario puede incluir un indicador de separación, que indica si la CU correspondiente al nodo está separada en sub-CU. Los elementos de sintaxis para una CU se pueden definir de manera recursiva y pueden depender de si la CU está separada en sub-CU. Si una CU no se separa adicionalmente, se denomina CU hoja. En la presente divulgación, cuatro sub-CU de una CU hoja también se denominarán CU hojas, incluso si no existe ninguna separación explícita de la CU hoja original. Por ejemplo, si una CU de tamaño 16x16 no se separa adicionalmente, las cuatro sub-CU de 8x8 también se denominarán CU hojas aunque la CU de 16x16 nunca se separara.

20 **[0056]** Una CU tiene un propósito similar a un macrobloque de la norma H.264, excepto porque una CU no tiene una distinción de tamaño. Por ejemplo, un bloque arbolado se puede separar en cuatro nodos hijos (también denominados sub-CU) y cada nodo hijo puede a su vez ser un nodo padre y separarse en otros cuatro nodos hijos. Un nodo hijo final, no separado, denominado nodo hoja del árbol cuaternario, comprende un nodo de codificación, también denominado una CU hoja. Los datos de sintaxis asociados con un flujo de bits codificado pueden definir 25 un número máximo de veces en que se puede separar un bloque arbolado, denominado máxima profundidad de CU, y también pueden definir un tamaño mínimo de los nodos de codificación. En consecuencia, un flujo de bits también puede definir una mínima unidad de codificación (SCU). La presente divulgación usa el término "bloque" para referirse a cualquiera de una CU, PU o TU, en el contexto de la HEVC, o a estructuras de datos similares en 30 el contexto de otras normas (por ejemplo, macrobloques y subbloques de las mismas en la H.264/AVC).

35 **[0057]** Una CU incluye un nodo de codificación y unidades de predicción (PU) y unidades de transformada (TU) 40 asociadas al nodo de codificación. Un tamaño de la CU corresponde a un tamaño del nodo de codificación y debe tener conformación cuadrada. El tamaño de la CU puede variar desde 8x8 píxeles hasta el tamaño del bloque arbolado, con un máximo de 64x64 píxeles o más. Cada CU puede contener una o más PU y una o más TU. Los 45 datos de sintaxis asociados con una CU pueden describir, por ejemplo, la división de la CU en una o más PU. Los modos de división pueden diferir entre si la CU está codificada en modo directo o de salto, codificada en modo de intrapredicción o codificada en modo de interpredicción. Las PU se pueden dividir para tener una conformación no cuadrada. Los datos de sintaxis asociados con una CU también pueden describir, por ejemplo, la división de la CU 50 en una o más TU de acuerdo con un árbol cuaternario. Una TU puede tener una conformación cuadrada o no cuadrada (por ejemplo, rectangular).

55 **[0058]** La norma HEVC permite transformaciones de acuerdo con las TU, que pueden ser diferentes para CU 60 diferentes. El tamaño de las TU se basa típicamente en el tamaño de las PU dentro de una CU dada definida para una LCU dividida, aunque puede que este no sea siempre el caso. Las TU son típicamente del mismo tamaño o más pequeñas que las PU. En algunos ejemplos, las muestras residuales correspondientes a una CU se pueden subdividir en unidades más pequeñas usando una estructura de árbol cuaternario conocida como "árbol cuaternario residual" (RQT). Los nodos hoja del RQT se pueden denominar unidades de transformada (TU). Los valores de 65 diferencias de píxeles asociados con las TU se pueden transformar para producir coeficientes de transformada, que se pueden cuantificar.

60 **[0059]** Una CU hoja puede incluir una o más unidades de predicción (PU). En general, una PU representa un área espacial correspondiente a la totalidad, o a una parte, de la CU correspondiente, y puede incluir datos para recuperar una muestra de referencia para la PU. Además, una PU incluye datos relacionados con la predicción. Por ejemplo, cuando la PU se codifica o descodifica en intramodo (es decir, se intrapredice o se codifica o descodifica por interpredicción), los datos para la PU se pueden incluir en un árbol cuaternario residual (RQT), que 65 puede incluir datos que describen un modo de intrapredicción para una TU correspondiente a la PU. Como otro

ejemplo, cuando la PU se codifica o descodifica en intermodo (es decir, se interpredice o se codifica o descodifica por interpredicción), la PU puede incluir datos que definen uno o más vectores de movimiento para la PU. Los datos que definen el vector de movimiento para una PU pueden describir, por ejemplo, una componente horizontal del vector de movimiento, una componente vertical del vector de movimiento, una resolución para el vector de movimiento (por ejemplo, precisión de un cuarto de píxel o precisión de un octavo de píxel), una imagen de referencia a la que apunta el vector de movimiento y/o una lista de imágenes de referencia (por ejemplo, la lista 0 o la lista 1) para el vector de movimiento.

**[0060]** Una CU hoja que tiene una o más PU también puede incluir una o más unidades de transformada (TU). Las unidades de transformada se pueden especificar usando un RQT (también denominado estructura de árbol cuaternario de TU), como se analiza anteriormente. Por ejemplo, un indicador de separación puede indicar si una CU hoja está separada en cuatro unidades de transformada. A continuación, cada unidad de transformada se puede separar adicionalmente en sub-TU adicionales. Cuando una TU no se separa adicionalmente, se puede denominar TU hoja. En general, para la intracodificación, todas las TU hojas que pertenecen a una CU hoja comparten el mismo modo de intrapredicción. Es decir, el mismo modo de intrapredicción se aplica en general para calcular valores predichos para todas las TU de una CU hoja. Para la intracodificación, un codificador de vídeo puede calcular un valor residual para cada TU hoja usando el modo de intrapredicción, como una diferencia entre la parte de la CU correspondiente a la TU y el bloque original. Una TU no está necesariamente limitada al tamaño de una PU. Por tanto, las TU pueden ser más grandes o más pequeñas que una PU. Para la intracodificación, una PU puede estar colocalizada con una TU hoja correspondiente para la misma CU. En algunos ejemplos, el tamaño máximo de una TU hoja puede corresponder al tamaño de la CU hoja correspondiente.

**[0061]** Además, las TU de las CU hojas también se pueden asociar con estructuras de datos de árbol cuaternario respectivas, denominadas árboles cuaternarios residuales (RQT). Es decir, una CU hoja puede incluir un árbol cuaternario que indica cómo se divide la CU hoja en las TU. El nodo raíz de un árbol cuaternario de una TU corresponde en general a una CU hoja, mientras que el nodo raíz de un árbol cuaternario de una CU corresponde en general a un bloque arbolado (o LCU). Las TU del RQT que no están separadas se denominan TU hojas. En general, la presente divulgación usa los términos CU y TU para hacer referencia a una CU hoja y a una TU hoja, respectivamente, a menos que se indique de otro modo.

**[0062]** 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 de sintaxis en una cabecera del GOP, una cabecera de una o más de las imágenes, o en otras ubicaciones, que describen un número de imágenes incluidas en el GOP. Cada fragmento de una imagen puede incluir datos de sintaxis de fragmento que describen un modo de codificación para el fragmento respectivo. El codificador de vídeo 20 funciona típicamente en bloques de vídeo dentro de fragmentos de vídeo individuales para codificar los datos de vídeo. Un bloque de vídeo puede corresponder a un nodo de codificación dentro de una CU. Los bloques de vídeo pueden tener tamaños fijos o variables y pueden diferir en tamaño de acuerdo con una norma de codificación especificada.

**[0063]** En un ejemplo, el HM admite la predicción en diversos tamaños de PU. Suponiendo que el tamaño de una CU particular es  $2Nx2N$ , el HM admite la intrapredicción en tamaños de PU de  $2Nx2N$  o  $NxN$ , y la interpredicción en tamaños de PU simétricas de  $2Nx2N$ ,  $2NxN$ ,  $Nx2N$  o  $NxN$ . El HM también admite la división asimétrica para la interpredicción en tamaños de PU de  $2NxN_U$ ,  $2NxN_D$ ,  $N_Lx2N$  y  $N_Rx2N$ . En la división asimétrica, una dirección de una CU no está dividida, mientras que la otra dirección está dividida en un 25 % y un 75 %. La parte de la CU correspondiente a la división de un 25 % se indica por una "n" seguida de una indicación de "arriba", "abajo", "izquierda" o "derecha". Por tanto, por ejemplo, " $2NxN_U$ " se refiere a una CU de  $2Nx2N$  que está dividida horizontalmente, con una PU de  $2Nx0,5N$  encima y una PU de  $2Nx1,5N$  debajo.

**[0064]** En la presente divulgación, " $NxN$ " y " $N$  por  $N$ " se pueden usar 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,  $16x16$  píxeles o  $16$  por  $16$  píxeles. En general, un bloque  $16x16$  tendrá  $16$  píxeles en una dirección vertical ( $y = 16$ ) y  $16$  píxeles en una dirección horizontal ( $x = 16$ ). Asimismo, un bloque  $NxN$  tiene, en general,  $N$  píxeles en una dirección vertical y  $N$  píxeles en una dirección horizontal, donde  $N$  representa un valor entero no negativo. Los píxeles de un bloque se pueden disponer en filas y columnas. Además, no es necesario que los bloques tengan necesariamente el mismo número de píxeles en la dirección horizontal que en la dirección vertical. Por ejemplo, los bloques pueden comprender  $NxM$  píxeles, donde  $M$  no es necesariamente igual a  $N$ .

**[0065]** Despues de la codificación intrapredictiva o interpredictiva (es decir, codificación o descodificación por intrapredicción o codificación o descodificación por interpredicción) usando las PU de una CU, el codificador de vídeo 20 puede calcular datos residuales para las TU de la CU. Las PU pueden comprender datos de sintaxis que describen un procedimiento o modo de generación de datos de píxeles predictivos en el dominio espacial (también denominado dominio de píxel) y las TU pueden comprender coeficientes en el dominio de transformada, después de la aplicación de una transformada, por ejemplo, una transformada de coseno discreta (DCT), una transformada de enteros, una transformada ondícula o una transformada conceptualmente similar, a los datos de vídeo residuales. Los datos residuales pueden corresponder a diferencias de píxeles entre píxeles de la imagen no

codificada y valores de predicción correspondientes a las PU. El codificador de vídeo 20 puede formar las TU, incluyendo los datos residuales para la CU y, a continuación, transformar las TU para producir coeficientes de transformada para la CU.

5      **[0066]** Despues de cualquier transformada para producir coeficientes de transformada, el codificador de vídeo 20 puede realizar la cuantificación de los coeficientes de transformada. La cuantificación se refiere, en general, a un procedimiento en el que coeficientes de transformada se cuantifican para reducir posiblemente la cantidad de datos usados para representar los coeficientes, proporcionando compresión adicional. El procedimiento de cuantificación puede reducir la profundidad de bits asociada con algunos, o con la totalidad, de los coeficientes.  
10     Por ejemplo, un valor de  $n$  bits puede redondearse a la baja a un valor de  $m$  bits durante la cuantificación, donde  $n$  es mayor que  $m$ .

15     **[0067]** Despues de la cuantización, el codificador de vídeo 20 puede explorar los coeficientes de transformada, produciendo un vector unidimensional a partir de la matriz bidimensional que incluye los coeficientes de transformada cuantificados. La exploración se puede diseñar para colocar los coeficientes de mayor energía (y, por lo tanto, de menor frecuencia) en la parte delantera de la matriz y para colocar los coeficientes de menor energía (y, por lo tanto, de mayor frecuencia) en la parte trasera de la matriz. En algunos ejemplos, el codificador de vídeo 20 puede utilizar un orden de exploración predefinido para explorar los coeficientes de transformada cuantificados, para producir un vector en serie que se puede codificar por entropía. En algunos ejemplos, el codificador de vídeo 20 puede realizar una exploración adaptativa. Despues de explorar los coeficientes de transformada cuantificados para formar un vector unidimensional, el codificador de vídeo 20 puede codificar por entropía el vector unidimensional, por ejemplo, de acuerdo con la codificación de longitud variable adaptativa al contexto (CAVLC), la codificación aritmética binaria adaptativa al contexto (CABAC), la codificación aritmética binaria adaptativa al contexto basada en la sintaxis (SBAC), la codificación por entropía por división en intervalos de probabilidad (PIPE) o con otra metodología de codificación por entropía. El codificador de vídeo 20 tambien puede codificar por entropía los elementos de sintaxis asociados con los datos de vídeo codificados, para su uso por el descodificador de vídeo 30 en la descodificación de los datos de vídeo.  
20  
25

30     **[0068]** Para realizar la CABAC, el codificador de vídeo 20 puede asignar un contexto, dentro de un modelo de contexto, a un símbolo que se va a transmitir. El contexto se puede referir, por ejemplo, a si los valores vecinos del símbolo son 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 se pueden construir de modo que los códigos relativamente más cortos correspondan a símbolos más probables, mientras que los códigos más largos correspondan a símbolos menos probables. De este modo, el uso de la VLC puede lograr un ahorro en bits con respecto, por ejemplo, al uso de palabras de código de igual longitud para cada símbolo que se va a transmitir. La determinación de la probabilidad se puede basar en un contexto asignado al símbolo.  
35

40     **[0069]** Ademas, el codificador de vídeo 20 puede enviar ademas datos de sintaxis, tales como datos de sintaxis basados en bloques, datos de sintaxis basados en tramas y datos de sintaxis basados en GOP, al descodificador de vídeo 30, por ejemplo, en una cabecera de trama, una cabecera de bloque, una cabecera de fragmento o una cabecera de GOP. Los datos de sintaxis de GOP pueden describir un número de tramas en el GOP respectivo, y los datos de sintaxis de trama pueden indicar un modo de codificación/predicción usado para codificar la trama correspondiente.  
45

50     **[0070]** Lo anterior describe técnicas de codificación de vídeo a modos de ejemplo en la norma HEVC. Estas técnicas de codificación de vídeo de ejemplo tambien son ampliables a técnicas de codificación de vídeo multivista, tales como las de MV-HEVC, 3D-HEVC, H.264/MVC y 3D-AVC. Para facilidad de ilustración y descripción solo, las técnicas descritas en la presente divulgación se describen con respecto a 3D-HEVC. Sin embargo, las técnicas de ejemplo pueden ser ampliables tambien a otras técnicas de codificación de vídeo multivista basadas en normas o distintas de normas.  
55

60     **[0071]** En general, en la codificación de vídeo multivista existen múltiples vistas, donde cada imagen correspondiente en cada vista incluye contenido de vídeo similar. Sin embargo, puede haber disparidad horizontal y/o vertical entre los bloques de vídeo en las imágenes correspondientes en las vistas diferentes. Es esta disparidad entre las imágenes lo que hace la experiencia de visualización en 3D. Como ejemplo, la pantalla 32 puede mostrar imágenes de dos o más vistas. Con gafas especializadas (u otras técnicas), el ojo izquierdo del espectador recibe imágenes de una de las vistas y no de las otras, y el ojo derecho del espectador recibe imágenes de una de las otras vistas y no de otra vista. El cerebro del espectador resuelve la disparidad entre las imágenes de estas vistas diferentes haciendo que el espectador experimente la visualización en 3D.  
65

60     **[0072]** En la codificación de vídeo multivista, para cada vista puede haber un componente de textura y un componente de profundidad correspondiente (denominado mapa de profundidad). El componente de textura incluye los datos de vídeo reales y se puede considerar como una imagen. El componente de profundidad indica la profundidad relativa de píxeles en el componente de profundidad correspondiente. El componente de profundidad puede no ser necesario en todos los ejemplos. En algunos ejemplos, una de las vistas puede incluir el componente de profundidad, pero es posible que otras vistas no.

5 [0073] Para la codificación de vídeo, la norma HEVC y otras técnicas de codificación de vídeo distintas de multivista permiten la interpredicción entre imágenes que se emiten o muestran en diferentes tiempos (por ejemplo, una imagen actual se interpredice con una imagen que se va a mostrar en un tiempo diferente). Además de permitir dicha interpredicción, las técnicas para la codificación de vídeo multivista permiten la predicción entre vistas, donde una imagen se interpredice con otra imagen en una vista diferente.

10 [0074] Para ayudar a comprender la codificación de vídeo multivista, lo siguiente proporciona alguna información adicional. Actualmente, un Equipo de colaboración conjunta en codificación de vídeo 3D (JCT-3C) de VCEG y MPEG está elaborando una norma de 3DV basada en la HEVC, para la que una parte de los trabajos de normalización incluye la normalización del códec de vídeo multivista basado en la HEVC (MV-HEVC) y otra parte, la codificación de vídeo 3D basada en la HEVC (3D-HEVC). Para la MV-HEVC, se debería garantizar que solo haya cambios de sintaxis de alto nivel (HLS) en ella, de modo que ningún módulo en el nivel de CU/PU en la HEVC necesite ser rediseñado y se pueda reutilizar por completo para la MV-HEVC. Para la 3D-HEVC, se pueden incluir y admitir nuevas herramientas de codificación, incluyendo aquellas a nivel de unidad de codificación/unidad de predicción, tanto para las vistas de textura como de profundidad. El software 3D-HTM más reciente para la 3D-HEVC se puede descargar desde el siguiente enlace: [3D-HTM, versión 7.0]: [https://hevc.hhi.fraunhofer.de/svn/svn\\_3DVCSoftware/tags/HTM-7.0/](https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-7.0/). La descripción del software de referencia más reciente, así como el borrador de trabajo de 3D-HEVC, están disponibles como sigue: Gerhard Tech, Krzysztof Wegner, Ying Chen, Sehoon Yea, "3D-HEVC Test Model 4", JCT3V-D1005\_spec\_v1, Equipo de colaboración conjunta en la elaboración de ampliación de codificación de vídeo 3D de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 4.<sup>a</sup> Conferencia: Incheon, Corea, 20-26 abr. 2013. Este documento se puede descargar desde el siguiente enlace: [http://phenix.it-sudparis.eu/jct2/doc\\_end\\_user/documents/4\\_Incheon/wg11/JCT3V-D1005-v1.zip](http://phenix.it-sudparis.eu/jct2/doc_end_user/documents/4_Incheon/wg11/JCT3V-D1005-v1.zip).

25 [0075] La MVC es una ampliación de H.264/AVC. La especificación MVC se analiza en resumen en las secciones y subsecciones de la presente divulgación a continuación. En la presente divulgación, cuando se detalla la codificación de vídeo multivista, la divulgación se refiere genéricamente a la codificación de vídeo donde existen múltiples vistas. Cuando se usa el acrónimo MVC, la divulgación se refiere a la norma de codificación de vídeo, que es una ampliación de H.264/AVC (a veces denominada H.264/MVC).

30 [0076] La FIG. 2 es un diagrama conceptual que ilustra una secuencia de codificación de vídeo multivista. Por ejemplo, un orden de descodificación de vídeo multivista típica (es decir, una orden de flujo de bits) se muestra en la FIG. 2. La disposición de orden de descodificación se denomina codificación de tiempo primero. Cada unidad de acceso (AU) se define para contener las imágenes codificadas de todas las vistas para un instante de tiempo de salida. Se debe tener en cuenta que el orden de descodificación de las unidades de acceso puede no ser idéntico al orden de salida o de visualización.

35 [0077] Por ejemplo, en la FIG. 2, S0-S7 se refieren a vistas diferentes, y SO se puede considerar como vista básica y las otras como vistas dependientes. En la FIG. 2, las imágenes identificadas como 0, 8, 16, 24... pertenecen a la vista S0, las imágenes identificadas como 1, 9, 17, 25... pertenecen a la vista S1, etc. Una AU incluye información de imagen para todas las imágenes de la misma o sustancialmente similar instancia de tiempo de salida. Por ejemplo, las imágenes 0-7 pertenecen a las vistas S0-S7, respectivamente, y se emiten aproximadamente en el tiempo T0.

40 [0078] La FIG. 3 es un diagrama conceptual que ilustra un patrón de predicción de codificación de vídeo multivista de ejemplo. Por ejemplo, una estructura típica de predicción de codificación de vídeo multivista (que incluye tanto la predicción entre imágenes dentro de cada vista como la predicción entre vistas) para la codificación de vídeo multivista se muestra en la FIG. 3, donde las predicciones se indican por flechas, usando el objeto al que se apunta el objeto desde el que se apunta como referencia de predicción.

45 [0079] En MVC, la predicción entre vistas se admite por la compensación de movimiento de disparidad, que usa la sintaxis de la compensación de movimiento de H.264/AVC, pero permite que se use una imagen de una vista diferente como imagen de referencia. Otras normas de codificación de vídeo multivista permiten de forma similar la predicción entre vistas. Por ejemplo, la MV-HEVC y la 3D-HEVC admiten la compensación de movimiento de disparidad usando la sintaxis de HEVC.

50 [0080] La codificación de dos vistas también se podría admitir por la codificación de vídeo multivista, y una de las ventajas de la codificación de vídeo multivista es que el codificador de vídeo 20 podría tomar más de dos vistas como entrada de vídeo 3D y el descodificador de vídeo 30 puede descodificar una representación multivista de este tipo. De este modo, un visualizador con un descodificador de codificación de vídeo multivista puede procesar contenido de vídeo 3D con más de dos vistas.

55 [0081] En la codificación de vídeo multivista, se permite la predicción entre vistas entre imágenes en la misma unidad de acceso (es decir, con la misma instancia de tiempo). Cuando se codifica una imagen en una de las vistas no básicas, se puede añadir una imagen a una lista de imágenes de referencia si está en una vista diferente pero

con una misma instancia de tiempo. Una imagen de referencia entre vistas se puede disponer en cualquier posición de una lista de imágenes de referencia, tal como cualquier imagen de referencia de interpredicción. Cuando se usa una imagen de referencia entre vistas para la compensación de movimiento, el vector de movimiento correspondiente se denomina "vector de movimiento de disparidad".

5 [0082] En la 3D-HEVC y la MV-HEVC, para mejorar además la eficacia de codificación, se han adoptado dos nuevas tecnologías, a saber, "predicción de movimiento entre vistas" y "predicción residual entre vistas" en el software de referencia más reciente. Para habilitar estas dos herramientas de codificación, la primera etapa es derivar un vector de disparidad. El vector de disparidad se usa para localizar el bloque correspondiente en la otra vista para el movimiento entre vistas/predicción residual o bien convertirse en un vector de movimiento de disparidad para la predicción de movimiento entre vistas.

10 15 [0083] La predicción de movimiento entre vistas y la predicción residual entre vistas son algunos ejemplos de herramientas de codificación de vídeo que utilizan un vector de disparidad para un bloque actual; sin embargo, puede haber herramientas de codificación de vídeo adicionales que usen un vector de disparidad, y las técnicas descritas en la presente divulgación no se limitan a estos ejemplos de herramientas de codificación de vídeo que utilizan un vector de disparidad. Las técnicas para determinar el vector de disparidad para un bloque actual y los usos de ejemplo del vector de disparidad, tales como en la predicción de movimiento entre vistas y la predicción residual entre vistas, se describen con más detalle a continuación.

20 25 [0084] En general, un vector de disparidad indica disparidad entre un bloque actual y un bloque correspondiente en otra vista. Por ejemplo, el contenido de vídeo del bloque actual y el contenido de vídeo del bloque correspondiente en la otra vista pueden ser similares, pero la localización de los bloques puede estar desplazada entre sí en las respectivas imágenes. El vector de disparidad proporciona una medida de este desplazamiento.

30 35 40 [0085] Un vector de disparidad y un vector de movimiento de disparidad no se deben confundir, ya que estos vectores son diferentes. Tanto un vector de disparidad como un vector de movimiento de disparidad se refieren a bloques en otras vistas; sin embargo, el propósito y uso del vector de disparidad y el vector de movimiento de disparidad es diferente. Un vector de movimiento de disparidad para un bloque actual se refiere a un bloque predictivo, donde el residuo entre el bloque predictivo y el bloque actual se transforma, cuantifica y señaliza como parte del flujo de bits. Un vector de disparidad, por otra por otra parte, se usa para diversas herramientas de codificación. Por ejemplo, a diferencia de un vector de movimiento de disparidad, que indica el vector de movimiento real para el bloque actual, el vector de disparidad se refiere a un bloque de referencia con información de movimiento que se puede usar posiblemente para determinar el vector de movimiento para el bloque actual. En otras palabras, el bloque al que se hace referencia por el vector de movimiento de disparidad es el bloque con valores de píxel que se restan de los valores de píxel del bloque actual. Por el contrario, es la información de movimiento del bloque al que se hace referencia por el vector de disparidad la que a continuación se usa para determinar el vector de movimiento para el bloque actual.

45 [0086] El bloque al que se hace referencia por el vector de disparidad se puede usar como un bloque de referencia para codificar o descodificar el bloque actual porque la información de movimiento del bloque de referencia se puede usar para codificar o descodificar el bloque actual. Además, el contenido de vídeo del bloque al que se hace referencia por el vector de disparidad puede ser similar al del bloque actual, y en este sentido se puede considerar como un bloque correspondiente al bloque actual. En consecuencia, en la presente divulgación, el bloque al que se hace referencia por el vector de disparidad se puede denominar bloque de referencia o bloque correspondiente.

50 55 [0087] Cabe destacar que en algunos casos, el vector de disparidad se convierte en un vector de movimiento de disparidad para el bloque actual. Sin embargo, esto solo puede suceder si se selecciona el vector de disparidad para que sea el vector de movimiento de disparidad. De este modo, el vector de disparidad es diferente del vector de movimiento de disparidad, excepto en el caso donde el vector de disparidad se selecciona específicamente para que sea el vector de movimiento de disparidad para el bloque actual. Además, en algunos ejemplos, en los casos donde el vector de disparidad se va a usar como un vector de movimiento de disparidad, se puede aplicar una función de recorte al vector de disparidad, donde la componente y del vector de disparidad se establece en cero y el vector resultante se usa como vector de movimiento de disparidad.

60 65 [0088] Las técnicas descritas en la presente divulgación están relacionadas con la identificación del bloque de referencia (por ejemplo, el bloque correspondiente) en una imagen de referencia en una vista de referencia en base al vector de disparidad determinado. Por ejemplo, qué bloque identifica el vector de disparidad se basa en el lugar desde donde comienza el vector de disparidad del bloque actual. Por ejemplo, un vector de disparidad incluye una posición de inicio que empieza en un píxel y una posición final que apunta a un píxel dentro de un bloque. El codificador de vídeo 20 y el descodificador de vídeo 30 pueden identificar el bloque de referencia para el bloque actual como el bloque que incluye (por ejemplo, cubre) el píxel al que apunta el vector de disparidad. El vector de disparidad para el bloque actual se referiría a diferentes píxeles para diferentes posiciones de inicio del vector de disparidad. Por lo tanto, en base a la posición de inicio del vector de disparidad, puede haber diferentes bloques de referencia posibles en la imagen de referencia en la vista de referencia a los que puede apuntar el vector de

disparidad.

**[0089]** Para identificar el bloque de referencia en base al vector de disparidad, la presente divulgación describe técnicas de ejemplo para determinar la posición final a la que se refiere el vector de disparidad. Debido a que la posición final a la que se refiere el vector de disparidad se basa en la posición de inicio del vector de disparidad, se puede considerar que las técnicas establecen la posición de inicio del vector de disparidad de modo que el vector de disparidad se refiere a la posición final determinada. En algunos ejemplos, la posición de inicio del vector de disparidad para el bloque actual puede estar dentro del bloque actual. En algunos ejemplos, la posición de inicio del vector de disparidad para el bloque actual puede ser externa al bloque actual.

**[0090]** Se debe entender que el propósito del vector de disparidad es identificar un bloque de referencia que se usa para diversas herramientas de codificación de vídeo para interpretar (incluyendo la predicción entre vistas) el bloque actual, tal como la predicción de movimiento entre vistas y la predicción residual entre vistas, como unos pocos ejemplos. El bloque de referencia que estadísticamente tiende a proporcionar una buena eficacia de codificación para codificar o descodificar el bloque actual puede requerir que el vector de disparidad para el bloque actual comience desde dentro del bloque actual o comience desde el exterior al bloque actual.

**[0091]** La FIG. 4 es un diagrama conceptual que ilustra posiciones de inicio de ejemplo para un vector de disparidad para identificar un bloque de referencia. Como se ilustra en la FIG. 4, la vista 0 incluye la imagen 34 y una vista diferente, la vista 1 incluye la imagen 36. La imagen 34 y la imagen 36 se pueden mostrar aproximadamente al mismo tiempo y, por consiguiente, pueden ser parte de la misma unidad de acceso.

**[0092]** En este ejemplo, la imagen 36 es la imagen actual en la vista actual (por ejemplo, la vista 1) y la imagen 34 es la imagen de referencia en la vista de referencia (por ejemplo, la vista 0). La imagen 36 incluye el bloque de vídeo 38. El bloque de vídeo 38 es el bloque de vídeo que se codifica o descodifica. Por ejemplo, desde la perspectiva de la codificación de vídeo, el bloque de vídeo 38 es el bloque actual que se está codificando por interprediccción por el codificador de vídeo 20. Desde la perspectiva de la descodificación de vídeo, el bloque de vídeo 38 es el bloque actual que se está descodificando por interprediccción por el descodificador de vídeo 30. En otras palabras, el ejemplo ilustrado en la FIG. 4 es aplicable a la situación donde el codificador de vídeo 20 codifica el bloque actual (por ejemplo, el bloque de vídeo 38), y es aplicable a la situación donde el descodificador de vídeo 30 descodifica el bloque actual (por ejemplo, el bloque de vídeo 38).

**[0093]** Debido a que la FIG. 4 ilustra un ejemplo aplicable tanto al codificador de vídeo 20 como al descodificador de vídeo 30, para facilitar la descripción, el ejemplo de la FIG. 4 se describe con respecto a un codificador de vídeo. Un codificador de vídeo, como se usa en la presente divulgación, es un término genérico para una unidad de procesamiento configurada para codificar o descodificar datos de vídeo. Por ejemplo, un ejemplo de un codificador de vídeo es el codificador de vídeo 20 y otro ejemplo de un codificador de vídeo es el descodificador de vídeo 30. Además, en la presente divulgación, el término código o codificación se usa para referirse genéricamente a codificar o codificación, respectivamente, o descodificar o descodificación, respectivamente. Por ejemplo, un codificador de vídeo que codifica o configurado para codificar se refiere genéricamente al codificador de vídeo 20 que codifica o configurado para codificar o al descodificador de vídeo 30 que descodifica o configurado para descodificar.

**[0094]** En el ejemplo ilustrado en la FIG. 4, un codificador de vídeo (por ejemplo, un codificador de vídeo 20 o un descodificador de vídeo 30) puede haber determinado un vector de disparidad (DV) para el bloque de vídeo 38. Por ejemplo, el codificador de vídeo puede haber utilizado técnicas de derivación de vector de disparidad basado en bloques vecinos (NBDV) (y posiblemente en combinación con técnicas de refinamiento) para derivar el vector de disparidad para el bloque de vídeo 38. La técnica de derivación de NBDV y los refinamientos para determinar el vector de disparidad para el bloque de vídeo 38 se describen con más detalle a continuación. Sin embargo, las técnicas descritas en la presente divulgación son aplicables a otros modos en los que determinar o derivar el vector de disparidad para el bloque de vídeo 38.

**[0095]** El codificador de vídeo también puede haber determinado que el vector de disparidad para el bloque de vídeo 38 se refiere a un bloque de referencia en la imagen 34. Por ejemplo, el elemento de sintaxis refViewIdx puede definir un valor de índice de la imagen de referencia, el codificador de vídeo puede determinar el valor de ViewIdx en base a refViewIdx. En este ejemplo, el valor de ViewIdx es igual a 0 para indicar la vista 0. El codificador de vídeo puede identificar el bloque de referencia en base a una localización (por ejemplo, un píxel en la imagen 34) a la que se refiere el vector de disparidad para el bloque de vídeo 38. Sin embargo, la localización en la imagen 34 a la que se refiere el vector de disparidad para el bloque de vídeo 38 se basa en una posición de inicio del vector de disparidad.

**[0096]** Por ejemplo, la localización (por ejemplo, coordenadas de píxeles en la imagen 34) donde se refiere el vector de disparidad se puede calcular en base a las componentes x e y del vector de disparidad y las coordenadas de una posición de inicio del vector de disparidad. Como un ejemplo, el codificador de vídeo puede sumar la coordenada x de la posición de inicio a la componente x del vector de disparidad para determinar la coordenada x de la localización en la imagen 34 a la que se refiere el vector de disparidad, y sumar la coordenada y de la posición

de inicio a la componente y del vector de disparidad para determinar la coordenada y de la localización en la imagen 34.

**[0097]** En algunos ejemplos, el codificador de vídeo puede desplazar o redondear las coordenadas del vector de disparidad para alinear el vector de disparidad para referirse a una precisión de enteros. Por ejemplo, sin el desplazamiento o redondeo, el vector de disparidad puede no referirse a un píxel, sino que puede apuntar a una localización entre píxeles (es decir, precisión de subpíxeles tal como precisión de un cuarto de píxel). Con el desplazamiento y/o el redondeo del vector de disparidad, el codificador de vídeo puede asegurar que el vector de disparidad se refiere a un píxel (es decir, precisión de enteros).

**[0098]** En algunos ejemplos, el codificador de vídeo puede determinar el bloque (por ejemplo, la unidad de predicción) que cubre la localización donde se refiere el vector de disparidad. El codificador de vídeo puede identificar el bloque determinado como el bloque de referencia que se usa para diversas herramientas de codificación de vídeo que se basan en el vector de disparidad, tal como predicción de movimiento entre vistas, predicción residual entre vistas o ejemplos donde el vector de disparidad se convierte en un vector de movimiento de disparidad como unos pocos ejemplos no limitantes de herramientas de codificación de vídeo que se basan en el vector de disparidad.

**[0099]** La FIG. 4 ilustra los vectores de disparidad DV1 y DV2. Los valores de los vectores de disparidad DV1 y DV2 son los mismos (es decir, la componente x y la componente y para DV1 y DV2 son las mismas), pero se ilustran por separado porque la posición de inicio de DV1 es diferente de la posición de inicio de DV2, y por lo tanto, la localización a la que se refieren DV1 y DV2 respectivamente son diferentes. Por ejemplo, el codificador de vídeo puede haber determinado el valor del vector de disparidad para el bloque de vídeo 38, y el valor del vector de disparidad es el mismo valor para DV1 así como para DV2.

**[0100]** En la FIG. 4, la posición de inicio del vector de disparidad DV1 es la posición 44 dentro del bloque de vídeo 38, y la posición de inicio del vector de disparidad DV2 es la posición 46 dentro del bloque de vídeo 38. La localización a la que se refiere el vector de disparidad DV1 es la localización C0 en la imagen de referencia 34 de la vista de referencia 0, y la localización a la que se refiere el vector de disparidad DV2 es la localización C3 en la imagen de referencia 34 de la vista de referencia 0.

**[0101]** Por ejemplo, el bloque de vídeo 38 se ilustra como un bloque 4x4, pero también puede ser de otros tamaños, tal como 8x8, 8x4, 4x8, 16x16, 16x8 u 8x16, como unos pocos ejemplos. Para bloques de dichos tamaños, es posible que no haya un único píxel central en el medio del bloque. Más bien, hay cuatro píxeles "centrales" candidatos dentro del bloque que forman un subbloque 2x2.

**[0102]** Como se ilustra en la FIG. 4, el centro del bloque de vídeo 38 está representado por el subbloque 2x2 central 48. Si el bloque de vídeo 38 fuera de un tamaño diferente a 4x4, de forma similar habría un subbloque 2x2 central. En la FIG. 4, la posición 44 es la posición del píxel superior izquierdo del subbloque 2x2 central 48, y la posición 46 es la posición del píxel inferior derecho del subbloque 2x2 central 48. La coordenada x del píxel superior izquierdo del subbloque 2x2 central 48 se puede calcular dividiendo el ancho del bloque de vídeo 38 entre dos y sumando el valor resultante a la coordenada x del píxel superior izquierdo del bloque de vídeo 38 y restando uno del valor resultante. La coordenada y del píxel superior izquierdo del subbloque 2x2 central 48 se puede calcular dividiendo la longitud del bloque de vídeo 38 entre dos y sumando el valor resultante a la coordenada y del píxel superior izquierdo del bloque de vídeo 38 y restando uno del valor resultante.

**[0103]** Las coordenadas del píxel superior derecho del subbloque 2x2 48 se pueden calcular sumando uno a la coordenada x del píxel superior izquierdo del subbloque 2x2 48 y manteniendo igual la coordenada y. Las coordenadas del píxel inferior izquierdo del subbloque 2x2 48 se pueden calcular sumando uno a la coordenada y del píxel superior izquierdo del subbloque 2x2 48 y manteniendo la coordenada x igual. Las coordenadas del píxel inferior derecho del subbloque 2x2 48 se pueden calcular sumando uno a la coordenada x del píxel superior izquierdo del subbloque 2x2 48 y sumando uno a la coordenada y del píxel superior izquierdo del subbloque 2x2 48.

**[0104]** Como ejemplo, supóngase que el píxel superior izquierdo del bloque de vídeo 38 está localizado en (5, 6) y, como se ilustra, el bloque de vídeo 38 es un bloque 4x4. En este ejemplo, la coordenada x del píxel superior izquierdo del subbloque 2x2 central 48 es  $5 + 4/2 - 1$ , que es igual a 6. La coordenada y del píxel superior izquierdo del subbloque 2x2 central 48 es  $6 + 4/2 - 1$ , que es igual a 7. En otras palabras, las coordenadas de la posición 44 son (6, 7). La coordenada x del píxel inferior derecho del subbloque 2x2 central 48 es  $6 + 1$ , que es 7, y la coordenada y del píxel inferior derecho del subbloque 2x2 central 48 es  $7 + 1$ , que es 8. En otras palabras, las coordenadas de la posición 46 son (7, 8).

**[0105]** Como se ilustra, si el vector de disparidad para el bloque de vídeo 38 comienza desde la posición 44 (por ejemplo, desde el píxel superior izquierdo del subbloque 2x2 central 48), entonces el vector de disparidad se refiere al píxel en la localización C0, como se ilustra por el vector de disparidad DV1. Si el vector de disparidad para el bloque de vídeo 38 comienza desde la posición 46 (por ejemplo, desde el píxel inferior derecho del subbloque 2x2

central 48), entonces el vector de disparidad se refiere al píxel en la localización C3, como se ilustra por el vector de disparidad DV2.

5 [0106] En algunos ejemplos, el codificador de vídeo puede determinar la localización C0 en la imagen de referencia 34 de la vista de referencia 0 y determinar el bloque que cubre la localización C0. Como se ilustra, el bloque que cubre la localización C0 en la FIG. 4 es el bloque 40. El codificador de vídeo puede identificar el bloque 40 como el bloque de referencia (por ejemplo, el bloque correspondiente) para el bloque de vídeo 38, y utilizar el bloque 40 para una o más herramientas de codificación de vídeo que se basan en la disparidad para el bloque de vídeo 38. Como un ejemplo, en la predicción de movimiento entre vistas, el codificador de vídeo puede utilizar la información de movimiento asociada con el bloque 40 para codificar el bloque de vídeo 38.

10 [0107] Como otro ejemplo de la utilización del bloque 40 para codificar el bloque de vídeo 38, el bloque 40 puede funcionar como un bloque predictivo para el bloque de vídeo 38. En otras palabras, el codificador de vídeo puede determinar un residuo entre los valores de píxel del bloque 40 y el bloque de vídeo 38. En estos ejemplos, el vector de disparidad se puede considerar como un vector de movimiento de disparidad para el bloque de vídeo 38. Además, en estos ejemplos, el codificador de vídeo puede almacenar una versión reconstruida del bloque 40 que el codificador de vídeo almacena para propósitos de interpredicción. En otras palabras, cuando el bloque 40 es un bloque predictivo, una versión reconstruida del bloque 40 funciona como bloque predictivo.

15 [0108] En este sentido, en algunos ejemplos, el codificador de vídeo puede utilizar la información de movimiento del bloque 40 para codificar el bloque de vídeo 38. En algunos ejemplos, el codificador de vídeo puede utilizar un bloque reconstruido 40 para codificar el bloque de vídeo 38. En cualquiera de estos ejemplos, el codificador de vídeo utiliza el bloque 40 para codificar por interpredicción el bloque de vídeo 38.

20 [0109] La técnica para determinar la localización C0 basada en el vector de disparidad para el bloque de vídeo 38 es como sigue. Indicar una localización de luma (xP, yP) de la muestra de luma superior izquierda de la unidad de predicción actual (por ejemplo, bloque de vídeo 38) en relación con la muestra de luma superior izquierda de la imagen actual (por ejemplo, imagen 36). Los valores nPSW y nPSH indican el ancho y la altura de la unidad de predicción actual (por ejemplo, el bloque de vídeo 38), respectivamente. Dado el índice de orden de vista de referencia refViewIdx, y un vector de disparidad mvDisp, la localización de luma de la capa de referencia (xRef, yRef) se deriva de:

$$xRef = Clip3( 0, PicWidthInSamples_L - 1, xP + ( ( nPSW - 1 ) >> 1 ) + ( ( mvDisp[ 0 ] + 2 ) >> 2 ) ) \quad (H-124)$$

$$yRef = Clip3( 0, PicHeightInSamples_L - 1, yP + ( ( nPSH - 1 ) >> 1 ) + ( ( mvDisp[ 1 ] + 2 ) >> 2 ) ) \quad (H-125)$$

35 [0110] En las ecuaciones anteriores, mvDisp [ 0 ] es igual a la componente x del vector de disparidad, mvDisp[ 1 ] es igual a la componente y del vector de disparidad y PicWidthInSamples\_L y PicHeightInSamples\_L definen la resolución de la imagen en la vista de referencia (igual que la vista actual). En las ecuaciones anteriores, xP + ((nPSW - 1) >> 1) e yP + ((nPSH - 1) >> 1) conjuntamente son iguales a la posición de inicio del vector de disparidad (es decir, xP + ((nPSW - 1) >> 1) es la coordenada x de la posición 44 e yP + ((nPSH - 1) >> 1) es la coordenada y de la posición 44). En las ecuaciones anteriores, xRef es la coordenada x para la localización dentro de la imagen 34 a la que apunta el vector de disparidad (por ejemplo, la coordenada x para la localización C0), e yRef es la coordenada y para la localización dentro de la imagen 34 a la que apunta el vector de disparidad (por ejemplo, la componente y para la localización C0). En la ecuación anterior, Clip3 es una función de recorte descrita con más detalle a continuación.

40 [0111] De este modo, las ecuaciones anteriores definen el modo de determinar el punto final del vector de disparidad DV1. En otras palabras, el codificador de vídeo puede establecer la posición de inicio del vector de disparidad igual a las coordenadas de la posición 44, y determinar las coordenadas de la localización C0 sumando las coordenadas respectivas de la posición de inicio al valor del vector de disparidad (por ejemplo, sumar la coordenada x de la posición 44 con la componente x del vector de disparidad, y la coordenada y de la posición 44 con la componente y del vector de disparidad).

45 [0112] El codificador de vídeo puede identificar el bloque de referencia (por ejemplo, el bloque correspondiente) como el bloque de vídeo que cubre las coordenadas xRef, yRef dentro de la imagen 34 (por ejemplo, que cubre la localización C0 dentro de la imagen 34). En el ejemplo ilustrado en la FIG. 4, el bloque 40 de la imagen de referencia 34 cubre la localización C0. En otras palabras, el bloque correspondiente se establece en la unidad de predicción que cubre la localización de luma (xRef, yRef) en el componente de vista con ViewIdx igual a refViewIdx. En este ejemplo, refViewIdx es el valor de índice usado para identificar la vista de referencia que incluye la imagen de

referencia, y ViewIdx es igual a la vista 0.

5 [0113] Sin embargo, puede haber bloques distintos del bloque 40 que estadísticamente tienden a proporcionar una mejor eficacia de codificación de vídeo para herramientas de codificación de vídeo que se basan en el vector de disparidad. En consecuencia, en las técnicas descritas en la presente divulgación, el codificador de vídeo puede identificar el bloque de referencia para el bloque de vídeo 38 como un bloque distinto del bloque que cubre la localización C0 en la imagen de referencia 34.

10 [0114] Como un ejemplo, el bloque que cubre el píxel localizado en la parte inferior derecha del píxel localizado en la localización C0 puede tender a proporcionar estadísticamente una mejor eficacia de codificación de vídeo que el bloque que cubre el píxel localizado en la localización C0. En la FIG. 4, la localización del píxel que está localizado en la parte inferior derecha de la localización C0 (por ejemplo, en la localización (1, 1) en relación con la localización C0) se denomina localización C3, y el bloque 42 cubre la localización C3.

15 [0115] Para determinar la localización de la localización C3 en la imagen de referencia 34, el codificador de vídeo puede establecer la posición de inicio del vector de disparidad en una posición diferente a la posición usada para determinar la localización C0 (por ejemplo, diferente de la posición 44). Por ejemplo, el codificador de vídeo puede establecer la posición de inicio del vector de disparidad igual a las coordenadas de la posición 46. El razonamiento para cambiar la posición de inicio del vector de disparidad es porque el valor del vector de disparidad sigue siendo el mismo, pero es necesaria una posición final diferente del vector de disparidad (por ejemplo, la localización C3, en lugar de la localización C0). Para cambiar la posición final del vector de disparidad, solo la posición de inicio puede cambiar porque el valor del vector de disparidad es el mismo.

20 [0116] Cabe destacar que puede ser posible cambiar el valor del vector de disparidad, en lugar de la posición de inicio, de modo que la posición final del vector de disparidad sea la localización C3, en lugar de la localización C0. Sin embargo, para propósitos de descripción, la divulgación se describe con respecto al cambio de la posición de inicio del vector de disparidad, en lugar del valor del vector de disparidad, de modo que el vector de disparidad se refiere a la localización C3 en lugar de la localización C0.

25 [0117] Para el ejemplo donde la posición de inicio de los vectores de disparidad cambia de modo que el vector de disparidad se refiere a la localización C3, la coordenada x para la posición de inicio (por ejemplo, la posición 46 en la FIG. 4) es  $xP + ((nPSW) >> 1)$  y la coordenada y para la posición de inicio (por ejemplo, la posición 46 en la FIG. 4) es  $yP + ((nPSH) >> 1)$ . La ecuación para determinar la localización de la localización C3 es como sigue.

$$xRef = Clip3( 0, PicWidthInSamples_L - 1, xP + ( nPSW >> 1 ) + ( ($$

35  $mvDisp[ 0 ] + 2 ) >> 2 ) ) ( H-124 )$

$$yRef = Clip3( 0, PicHeightInSamples_L - 1, yP + ( nPSH >> 1 ) + ( ($$

$mvDisp[ 1 ] + 2 ) >> 2 ) ) ( H-125 )$

40 [0118] En las ecuaciones inmediatamente anteriores, xRef es la coordenada x para la localización dentro de la imagen 34 a la que apunta el vector de disparidad (por ejemplo, la coordenada x para la localización C3), e yRef es la coordenada y para la localización dentro de la imagen 34 a la que apunta el vector de disparidad (por ejemplo, la componente y para la localización C3). En las ecuaciones anteriores,  $xP + (nPSW >> 1)$  e  $yP + (nPSH >> 1)$  conjuntamente son iguales a la posición de inicio del vector de disparidad (es decir,  $xP + (nPSW >> 1)$  es la coordenada x de la posición 46 e  $yP + (nPSH >> 1)$  es la coordenada y de la posición 46). En otras palabras, en las ecuaciones anteriores, las coordenadas del píxel inferior derecho en el subbloque 2x2 central 48 dentro del bloque actual 38 son iguales ( $(xP + (nPSW >> 1)), (yP + (nPSH >> 1))$ ), donde  $yP$  es igual a una coordenada y de una esquina superior izquierda del bloque actual 38, donde  $xP$  es igual a una coordenada x de una esquina superior izquierda del bloque actual 38, donde  $nPSW$  es igual a un ancho del bloque actual 38, y donde  $nPSH$  es igual a la altura del bloque actual 38.

45 [0119] De este modo, las ecuaciones anteriores definen el modo de determinar el punto final del vector de disparidad DV2. Por ejemplo, el codificador de vídeo puede establecer la posición de inicio del vector de disparidad igual a las coordenadas de la posición 46 y determinar las coordenadas de la localización C3 sumando las coordenadas respectivas de la posición de inicio al valor del vector de disparidad (por ejemplo, sumar la coordenada x de la posición 46 con la componente x del vector de disparidad, y la coordenada y de la posición 46 con la componente y del vector de disparidad). Por ejemplo, el codificador de vídeo puede determinar una coordenada x y una coordenada y para la localización en la imagen de referencia 34 a la que se refiere el vector de disparidad DV2 en base al vector de disparidad comenzando desde un píxel inferior derecho en el subbloque 2x2 central 48 dentro del bloque de vídeo 38 actual.

50 [0120] En las ecuaciones anteriores, el codificador de vídeo puede aplicar una operación de recorte para determinar xRef e yRef. Dicho recorte puede no ser necesario en todos los ejemplos. Las ecuaciones para xRef e

yRef sin la función de recorte pueden ser como sigue.

xRef = xP + (( nPSW ) >> 1 ) + (( mvDisp[ 0 ] + 2 ) >> 2 ),

yRef = yP + (( nPSH) >> 1 ) + (( mvDisp[ 1 ] + 2 ) >> 2 )).

**[0121]** En las técnicas descritas en la presente divulgación, el codificador de vídeo puede determinar un bloque en una imagen de referencia en una vista de referencia en base a una localización en la imagen de referencia a la que se refiere el vector de disparidad cuando el vector de disparidad comienza desde un píxel inferior derecho en un subbloque 2x2 central dentro del bloque actual (por ejemplo, en base al vector de disparidad comenzando desde un píxel inferior derecho en un subbloque 2x2 central dentro del bloque actual). Por ejemplo, como se ilustra en la FIG. 4, el codificador de vídeo puede determinar un bloque 4x4 (por ejemplo, el bloque 42) que cubre la localización C3 en la imagen de referencia 34 a la que se refiere el vector de disparidad DV2 en base al vector de disparidad comenzando desde la posición 46 (por ejemplo, cuando el vector de disparidad comienza desde la posición 46). El codificador de vídeo puede determinar las coordenadas x e y de la localización C3 a la que se refiere el vector de disparidad usando las ecuaciones para xRef e yRef proporcionadas anteriormente, y determinar el bloque que cubre las coordenadas x e y determinadas.

**[0122]** Es la información de movimiento del bloque 42 o los valores de píxel del bloque 42 lo que el codificador de vídeo puede usar a continuación para codificar por interprediccción (por ejemplo, codificar o descodificar) el bloque actual 38 usando herramientas de codificación de vídeo que se basan en el vector de disparidad. Como un ejemplo, el codificador de vídeo puede insertar la información de movimiento del bloque 42 en una lista de predictores de vector de movimiento candidatos. Si la información de movimiento del bloque 42 se selecciona de la misma lista de predictores de vector de movimiento candidatos que la información de movimiento que el codificador de vídeo va a usar para predecir el vector de movimiento para el bloque 38 actual, entonces el codificador de vídeo puede codificar por interprediccción el bloque actual en base a la información de movimiento del bloque determinado.

**[0123]** Como otro ejemplo, el codificador de vídeo puede utilizar el bloque 42 como bloque predictivo. Por ejemplo, el codificador de vídeo almacena una versión reconstruida del bloque 42. Por ejemplo, durante la codificación de vídeo, el codificador de vídeo 20 incluye una ruta de realimentación en la que el codificador de vídeo 20 reconstruye los bloques codificados, y almacena los bloques reconstruidos en la memoria de modo que se puede usar el bloque 42 para codificar los bloques posteriores. Durante la descodificación de vídeo, el descodificador de vídeo 30 reconstruye el bloque 42 y almacena el bloque reconstruido 42 en la memoria de modo que se puede usar el bloque 42 para descodificar los bloques posteriores. En algunos casos, el codificador de vídeo determina una diferencia entre los valores de píxel del bloque de vídeo 38 y el bloque 42, tal como en los ejemplos donde el bloque 42 funciona como un bloque predictivo para el bloque de vídeo 38. En los ejemplos donde una reconstrucción del bloque 42 funciona como un bloque predictivo, el vector de disparidad se puede considerar como un vector de movimiento de disparidad para el bloque de vídeo 38.

**[0124]** En otras palabras, el codificador de vídeo puede codificar por interprediccion el bloque actual en base al bloque determinado. Como un ejemplo, el codificador de vídeo puede codificar por interprediccion el bloque actual en base a la informacion de movimiento del bloque determinado si el bloque determinado se va a usar para codificar por interprediccion el bloque actual. Como otro ejemplo, el codificador de vídeo codifica por interprediccion el bloque actual en base a valores de píxel de una reconstrucción del bloque determinado.

**[0125]** En algunos casos, xRef e yRef se pueden recortar para asegurar que el bloque al que se hace referencia por el vector de disparidad todavía está dentro de la imagen de referencia 34. Además, cuando el vector de disparidad tiene una precisión de  $\frac{1}{4}$  de píxel, las operaciones de desplazamiento ( $mvDisp[i] + 2 \gg 2$ ) se pueden reemplazar con  $mvDisp[i] \& 1$ , donde  $i$  es igual a 0 o 1.

**[0126]** En este ejemplo, el codificador de vídeo puede identificar el bloque de referencia (por ejemplo, el bloque correspondiente) como el bloque de vídeo que cubre las coordenadas xRef, yRef dentro de la imagen 34 (por ejemplo, que cubre la localización C3 dentro de la imagen 34). En el ejemplo ilustrado en la FIG. 4, el bloque 42 de la imagen de referencia 34 cubre la localización C3. En otras palabras, el bloque correspondiente se establece en la unidad de predicción que cubre la localización de luma (xRef, yRef) en el componente de vista con ViewIdx igual a refViewIdx. En este ejemplo, refViewIdx es el valor de índice usado para identificar la vista de referencia que incluye la imagen de referencia, y ViewIdx es igual a la vista 0.

**[0127]** Como se puede ver en la FIG. 4, la localización a la que se refiere el vector de disparidad afecta qué bloque identifica el codificador de vídeo como bloque de referencia, y la localización a la que se refiere el vector de disparidad se basa en la posición de inicio del vector de disparidad. Por lo tanto, la posición de inicio del vector de disparidad afecta a qué bloque en la imagen de referencia 34 se identifica como bloque de referencia (por ejemplo, el bloque 40 o el bloque 42).

5 [0128] La información de movimiento del bloque 42 o los valores de píxel de una reconstrucción del bloque 42 pueden tender estadísticamente a proporcionar mejores eficacias de codificación de vídeo para herramientas de codificación de vídeo que se basan en el vector de disparidad que la información de movimiento del bloque 40 o los valores de píxel de una reconstrucción del bloque 40. En consecuencia, en algunos ejemplos, el codificador de

10 vídeo puede identificar el bloque que cubre la localización C3 (por ejemplo, el bloque 42) como el bloque de referencia para el bloque actual (por ejemplo, el bloque de vídeo 38).

15 [0129] En algunos ejemplos, en lugar de identificar el bloque que cubre la localización C3 en la imagen de referencia 34 de la vista de referencia 0 como bloque de referencia para el bloque de vídeo 38, el codificador de vídeo puede identificar el bloque que cubre la localización C1 o la localización C2 en la imagen de referencia 34 de la vista de referencia 0 como bloque de referencia para el bloque de vídeo 38. Por ejemplo, la FIG. 4 ilustra un píxel localizado en la localización C1 que está a la derecha del píxel localizado en C0 (por ejemplo, en la localización (1, 0) en relación con la localización C0), y un píxel localizado en la localización C2 que está debajo del píxel localizado en C0 (por ejemplo, en la localización (0, 1) en relación con la localización C0). En este caso, el píxel localizado en la localización C3 está localizado (1, 1) en relación con la localización C0.

20 [0130] El codificador de vídeo puede implementar técnicas similares a las descritas anteriormente para determinar la localización de la localización C1 o la localización C2 en la imagen de referencia 34. Por ejemplo, el codificador de vídeo puede cambiar la posición de inicio del vector de disparidad de modo que el vector de disparidad se refiere a la localización C1 (es decir, la posición de inicio es el píxel superior derecho en el subbloque central 2x2 48) o la localización C2 (es decir, la posición de inicio es el píxel inferior izquierdo en el subbloque central 2x2 48). A continuación, el codificador de vídeo puede identificar el bloque que cubre la localización C1 o C2 como bloque de referencia para el bloque actual (por ejemplo, para el bloque de vídeo 38).

25 [0131] En algunos ejemplos, el codificador de vídeo puede comprobar múltiples bloques que cubren diferentes localizaciones en la imagen de referencia 34 para identificar el bloque de referencia para el bloque de vídeo 38. Por ejemplo, el codificador de vídeo puede determinar en primer lugar el bloque que cubre la localización C3 (por ejemplo, el bloque 42). A continuación, el codificador de vídeo puede determinar si la información de movimiento del bloque 42 está disponible para una o más herramientas de codificación de vídeo que se basan en el vector de disparidad.

30 [0132] Por ejemplo, si el bloque 42 se codifica por intrapredicción, entonces no hay información de movimiento para el bloque 42. En otro ejemplo, si el bloque 42 se codifica por predicción entre vistas, entonces la información de movimiento del bloque 42 puede no ser utilizable para codificar el bloque de vídeo 38 actual usando una o más herramientas de codificación de vídeo que se basan en el vector de disparidad. Además, en otro ejemplo, si el bloque 42 se interpreta, pero la imagen en la vista actual que está localizada en la misma unidad de acceso que la imagen de referencia a la que se refiere el vector de movimiento del bloque 42 no se identifica en la(s) lista(s) de imágenes de referencia del bloque de vídeo 38, entonces la información de movimiento del bloque 42 puede no ser utilizable para codificar el bloque de vídeo 38 actual usando una o más herramientas de codificación de vídeo que se basan en el vector de disparidad.

35 [0133] Si el codificador de vídeo determina que la información de movimiento para el bloque 42 está disponible para las una o más herramientas de codificación de vídeo que se basan en el vector de disparidad, entonces el codificador de vídeo puede identificar el bloque 42 como bloque de referencia para el bloque de vídeo 38. Si el codificador de vídeo determina que la información de movimiento para el bloque 42 no está disponible para las una o más herramientas de codificación de vídeo que se basan en el vector de disparidad, el codificador de vídeo puede determinar que no hay ningún bloque de referencia disponible para el bloque de vídeo 38. En algunos ejemplos, en lugar de determinar que no hay ningún bloque de referencia para el bloque de vídeo 38, el codificador de vídeo puede determinar el bloque que cubre la localización C0 (por ejemplo, el bloque 40), y determinar si la información de movimiento para el bloque 40 está disponible para las una o más herramientas de codificación de vídeo que se basan en el vector de disparidad. Si el codificador de vídeo determina que la información de movimiento para el bloque 40 está disponible para las una o más herramientas de codificación de vídeo que se basan en el vector de disparidad, el codificador de vídeo puede determinar que el bloque 40 es el bloque de referencia para el bloque de vídeo 38 actual. De otro modo, el codificador de vídeo puede determinar que no hay ningún bloque de referencia disponible para el bloque de vídeo 38.

40 [0134] En algunos ejemplos, el codificador de vídeo puede realizar las etapas opuestas. Por ejemplo, el codificador de vídeo puede determinar en primer lugar si la información de movimiento del bloque 40 está disponible para las una o más herramientas de codificación de vídeo que se basan en el vector de disparidad, y determinar que el bloque 40 es el bloque de referencia para el bloque de vídeo 38 si la información de movimiento está disponible. Si la información de movimiento no está disponible, entonces el codificador de vídeo puede determinar si la información de movimiento para el bloque 42 está disponible y, de ser así, determinar que el bloque 42 es el bloque de referencia para el bloque de vídeo 38 actual. Si la información de movimiento para el bloque 42 no está disponible, entonces el codificador de vídeo puede determinar que no existe ningún bloque de referencia para el bloque de vídeo 38.

5 [0135] No es necesario que el codificador de vídeo interrumpa necesariamente la comprobación del bloque de referencia después de dos bloques (por ejemplo, después de los bloques 40 y 42, en cualquier orden). En algunos ejemplos, el codificador de vídeo puede comprobar los bloques 40 y 42 (en cualquier orden), y si ninguno de los bloques está disponible para las herramientas de codificación de vídeo que se basan en el vector de disparidad, el codificador de vídeo puede determinar el bloque que cubre la localización C1 o el bloque que cubre la localización C2 (en cualquier orden) y determinar si cualquiera de estos bloques está disponible. En general, las diversas permutaciones de los órdenes de comprobación de los bloques que cubren las localizaciones C0, C1, C2 y C3 son posibles, y están contempladas en la presente divulgación.

10 10 [0136] Los ejemplos anteriores describen la identificación de un bloque de referencia en base al bloque que cubre la localización a la que se refiere el vector de disparidad (por ejemplo, los puntos). Sin embargo, las técnicas descritas en la presente divulgación no están limitadas así. En algunos ejemplos, el codificador de vídeo puede determinar la localización de un bloque en base al vector de disparidad, y determinar si el bloque es el bloque de referencia para el bloque de vídeo 38.

15 15 [0137] Como un ejemplo, el codificador de vídeo puede identificar el bloque "inferior derecho" en la imagen de referencia 34, denominada BR0. Por ejemplo, si el vector de disparidad del bloque de vídeo 38 actual comenzó desde el bloque que está en la parte inferior derecha del bloque de vídeo 38 actual, la localización a la que apunta el vector de disparidad en la imagen de referencia 34 es la esquina superior izquierda del bloque BR0. En este ejemplo, la posición de inicio del vector de disparidad para el bloque de vídeo 38 es externa al bloque de vídeo 38. La coordenada x para el bloque BR0 es  $xP + (nPSW - 1) + ((mvDisp[ 0 ] + 2) >> 2)$ , y la coordenada y para el bloque BR0 es  $yP + (nPSH-1) + ((mvDisp[ 1 ] + 2) >> 2)$ . Además, en este ejemplo, la coordenada x de la posición de inicio del vector de disparidad es  $xP + (nPSW - 1)$ , y la coordenada y de la posición de inicio del vector de disparidad es  $yP + (nPSH-1)$ . La posición de inicio para el vector de disparidad que se refiere al bloque BR0 es diferente a la posición de inicio del vector de disparidad que se refiere al bloque 40 o al bloque 42, y puede ser externa al bloque de vídeo 38.

20 30 [0138] En algunos ejemplos, en lugar de comprobar los bloques 42, 40 o los bloques que cubren las localizaciones C1 y C2, el codificador de vídeo puede determinar si la información de movimiento para el bloque BR0 está disponible para las herramientas de codificación de vídeo que se basan en el vector de disparidad. Si está disponible, el codificador de vídeo puede identificar el bloque BR0 como bloque de referencia y, de otro modo, determinar que no existe ningún bloque de referencia. Sin embargo, en algunos ejemplos, también puede haber otros bloques para comprobar. Estos otros bloques de la imagen de referencia 34 están etiquetados como bloque BR1 y bloque BR2. El bloque BR1 está localizado (1, 1) en relación con el bloque BR0, y el bloque BR2 está localizado (2, 2) en relación con el bloque BR0. El codificador de vídeo puede determinar de forma similar la localización de los bloques BR1 y BR2 como se determina para el bloque BR0 (por ejemplo, cambiando la posición de inicio del vector de disparidad de modo que el vector de disparidad se refiera al bloque BR1 o al bloque BR2).

35 40 [0139] Como modos de ejemplo adicionales para identificar el bloque de referencia, el codificador de vídeo puede, en primer lugar, desplazar y cuantificar la posición de la esquina izquierda superior del bloque al que se hace referencia por el vector de disparidad si el vector de disparidad comienza desde la esquina superior izquierda del bloque de vídeo 38. Por ejemplo, el codificador de vídeo puede implementar la siguiente ecuación:  $((xP + (mvDisp[ 0 ] + 2) >> 2) + 2) >> 2, (yP + (mvDisp[ 1 ] + 2) >> 2) + 2) >> 2$ . Después de eso, el codificador de vídeo puede aplicar directamente un desplazamiento de  $(nPSW/2, nPSH/2)$ ,  $(nPSW, nPSH)$  para identificar un bloque 4x4. Otros bloques potenciales se pueden basar en un desplazamiento de  $(nPSW/2-4, nPSH/2-4)$ ,  $(nPSW/2+4, nPSH/2+4)$ ,  $(nPSW-4, nPSH-4)$  y  $(nPSW+4, nPSH+4)$ . Después de aplicar un desplazamiento, el codificador de vídeo puede identificar como bloque de referencia un bloque 4x4 usando la posición de muestra desplazada como la posición superior izquierda.

50 55 [0140] Se debe entender que en las técnicas descritas anteriormente son posibles diversas permutaciones. Por ejemplo, una orden de comprobación puede incluir el bloque que cubre la localización C3, a continuación el bloque que cubre la localización C0, a continuación el bloque que cubre la localización C1, a continuación el bloque que cubre C2, a continuación el bloque BR0, a continuación el bloque BR1, a continuación el bloque BR2, y a continuación la aplicación de la técnica de desplazamiento para identificar el bloque de referencia, e interrumpir la búsqueda del bloque de referencia si ninguno de los bloques incluye información de movimiento disponible para las una o más herramientas de codificación de vídeo. Sin embargo, las técnicas no están limitadas así. Otras órdenes de comprobación pueden ser posibles, y contempladas en la presente divulgación.

60 65 [0141] Además, no es necesario que el codificador de vídeo necesariamente compruebe todos los bloques posibles, y puede comprobar solo un subconjunto de los bloques (por ejemplo, solo un bloque, o solo dos bloques, etc.). Por ejemplo, el codificador de vídeo puede comenzar con el bloque 42, y si el bloque 42 no está disponible, determinar que no está disponible ningún bloque de referencia para el bloque de vídeo 38. En algunos ejemplos, se pueden obtener más ganancias de codificación comprobando sólo el bloque 42, e interrumpiendo si el bloque 42 no está disponible. El motivo puede ser que la comprobación de bloques adicionales requiere un ancho de banda de memoria adicional, e incluso si hay alguna ganancia de codificación de vídeo al identificar un bloque con información de movimiento que se puede usar como bloque de referencia si la información de movimiento para el

bloque 42 no está disponible, el coste de las solicitudes de memoria adicional puede superar el beneficio.

**[0142]** El codificador de vídeo 20 y el descodificador de vídeo 30 se pueden configurar para realizar implícitamente una o una combinación de dos o más de las técnicas de ejemplo descritas anteriormente de modo que el codificador de vídeo 20 y el descodificador de vídeo 30 identifican el mismo bloque como bloque de referencia para el bloque de vídeo 38. Por ejemplo, es posible que el codificador de vídeo 20 no necesite señalizar información que indica qué bloque es el bloque de referencia en la imagen de referencia 34, y es posible que el descodificador de vídeo 30 no necesite recibir información que indica qué bloque es el bloque de referencia en la imagen de referencia 34.

**[0143]** Sin embargo, en algunos ejemplos, el codificador de vídeo 20 puede comprobar diversos bloques y determinar qué bloque proporciona las ganancias de codificación máximas si se identifica como bloque de referencia. Por ejemplo, es posible que el codificador de vídeo 20 no deje de comprobar bloques después de determinar que uno de los bloques de la imagen de referencia 34 incluye información de movimiento disponible para una o más herramientas de codificación de vídeo que se basan en el vector de disparidad, y puede determinar una pluralidad de bloques que incluyen información de movimiento disponible para una o más herramientas de codificación de vídeo que se basan en el vector de disparidad. A continuación, el codificador de vídeo 20 puede determinar cuál de estos bloques proporciona las ganancias de codificación máximas, y la información de señal que indica qué bloque es el bloque de referencia. El descodificador de vídeo 30 puede recibir la información que indica qué bloque es el bloque de referencia, e identificar el bloque de referencia en base a la información recibida.

**[0144]** Además, aunque las técnicas se han descrito con respecto a un vector de disparidad, las técnicas no están limitadas así. Por ejemplo, las técnicas se pueden generalizar a cualquier supuesto cuando es necesario identificar un elemento de sintaxis o una variable asociada con un bloque.

**[0145]** De acuerdo con la presente divulgación, para las herramientas de codificación de vídeo que se basan en un vector de disparidad, tales como la predicción de movimiento entre vistas, sin importar la textura o la profundidad (descritas con más detalle a continuación), es posible que el codificador de vídeo necesite identificar un bloque en un vista de referencia identificando, en primer lugar, un píxel. Algunas técnicas existentes pueden no ser lo suficientemente exactas.

**[0146]** Para abordar estos problemas, las técnicas descritas en la presente divulgación se pueden resumir como sigue. Por ejemplo, la presente divulgación describe técnicas para identificar diferentes píxeles durante la identificación de un bloque de referencia en la vista de referencia. Las técnicas pueden ser aplicables a la predicción de movimiento entre vistas para una vista de textura o una vista de profundidad, o en general a diversas herramientas de codificación de vídeo que se basan en un vector de disparidad.

**[0147]** En las técnicas descritas en la presente divulgación, indicar  $(xP, yP)$  para que sea la posición de la esquina superior izquierda de la PU actual y un vector de disparidad  $mvDisp$  (con una exactitud de 1/4 de pixel) para la PU actual. Indicar la muestra central identificada por este vector de disparidad como  $C0$ , con una coordinación relativa de  $(xP + ((nPSh - 1) \gg 1) + ((mvDisp[ 0 ] + 2) \gg 2), yP + ((nPSh - 1) \gg 1) + ((mvDisp[ 1 ] + 2) \gg 2))$ . Indicar los otros tres píxeles contiguos a  $C0$  como  $C1, C2, C3$  con una posición relativa con respecto a  $C0$  como  $(1, 0), (0, 1)$  y  $(1, 1)$ . Indicar la muestra de la parte inferior derecha con una coordinación de  $(xP + (nPSh - 1) + ((mvDisp[ 0 ] + 2) \gg 2), yP + (nPSh - 1) + ((mvDisp[ 1 ] + 2) \gg 2))$  como  $BR0$ .  $BR1$  y  $BR2$  son dos muestras con una posición relativa con respecto a  $BR0$  como  $(1, 1)$  y  $(2, 2)$ .

**[0148]** Se debe tener en cuenta que para cualquiera de las muestras anteriores, la componente horizontal y vertical se puede recortar para asegurarse de que la muestra todavía está dentro de una imagen. Se debe tener en cuenta que cuando el vector de disparidad tiene una exactitud de entero, las operaciones de desplazamiento  $((mvDisp[ i ] + 2) \gg 2)$  se podrían simplemente reemplazar por  $mvDisp[ i ]$ .

**[0149]** En un ejemplo, en lugar de identificar el bloque identificando el píxel  $C0$ , se utiliza el píxel  $C3$  y el bloque se identifica como el bloque 4x4/unidad de predicción que cubre el píxel  $C3$ . La posición  $C3$  se calcula como sigue.

$$\begin{aligned} xRef &= Clip3( 0, PicWidthInSamplesL - 1, xP + ( ( nPSW ) \gg 1 ) + ( ( mvDisp[ 0 ] \\ &\quad + 2 ) \gg 2 ) ) ( H-124 ) \\ yRef &= Clip3( 0, PicHeightInSamplesL - 1, yP + ( ( nPSH ) \gg 1 ) + ( ( mvDisp[ 1 ] \\ &\quad + 2 ) \gg 2 ) ) ( H-125 ) \end{aligned}$$

donde  $xRef$  representa la coordenada x e  $yRef$  representa la coordenada y de una localización en la imagen de referencia, y el bloque que cubre esta localización es el bloque de referencia.

**[0150]** En algunos ejemplos, se puede usar  $C1$  o  $C2$ . En algunos ejemplos, se pueden usar  $BR0$ ,  $BR1$  o  $BR2$ . En algunos ejemplos, se usa  $C0$  en primer lugar, además, cuando el bloque 0 que cubre  $C0$  no produce un candidato

de fusión disponible, el bloque 3 que cubre C3, si es diferente del bloque 0, se usa para producir un candidato de fusión disponible (descrito con más detalle a continuación). En algunos ejemplos, se usa C3 en primer lugar. Además, cuando el bloque 3 que cubre C3 no produce un candidato de fusión disponible, el bloque 0 que cubre C0, si es diferente del bloque 0, se usa para producir un candidato de fusión disponible.

5 [0151] En algunos ejemplos, BR0, BR1 o BR2 se usan para identificar un bloque 4x4 que cubre el BR0, a saber, el bloque B0, un bloque 4x4 que cubre el BR1, a saber, el bloque B1, y el bloque 4x4 que cubre BR2, a saber, el bloque B2, del que se puede derivar un candidato de fusión. En algunos ejemplos, cuando el candidato de fusión del bloque 0 o el bloque 3, o del bloque 0 o bien del bloque 3 no está disponible, se usan más candidatos de fusión de uno o más bloques del bloque B0, el bloque B1 y el bloque B2.

10 [0152] En algunos ejemplos, la posición de la esquina izquierda superior en primer lugar se desplaza y se cuantifica para que esté alineada como 4x4, por ejemplo, como sigue: calcular ((xP+(mvDisp[ 0 ] + 2 )>> 2 )+2)>>2, (yP+(mvDisp[ 1 ] + 2 )>> 2 )+2)>>2; después de eso, se puede aplicar directamente un desplazamiento de (nPSPW/2, nPSH/2), (nPSPW, nPSH) para identificar un bloque 4x4. Otros bloques potenciales se pueden basar en un desplazamiento de (nPSPW/2-4, nPSH/2-4), (nPSPW/2+4, nPSH/2+4), (nPSPW-4, nPSH-4) y (nPSPW+4, nPSH+4). Despues de aplicar un desplazamiento, se identifica un bloque 4x4 usando la posición de muestra desplazada como la posición superior izquierda.

15 [0153] Como se describe anteriormente, las técnicas descritas en la presente divulgación están relacionadas con normas de codificación de vídeo tales como 3D-HEVC. Lo siguiente proporciona algo de contexto para HEVC.

20 [0154] Las listas de imágenes de referencia se utilizan para identificar qué imágenes pueden ser imágenes de referencia para la imagen actual. Por ejemplo, cada una de las listas de imágenes de referencia incluye una pluralidad de entradas que son direccionables por su índice y que identifican una imagen de referencia. El codificador de vídeo puede construir una lista de imágenes de referencia para una imagen P y dos listas de imágenes de referencia para una imagen B. Las listas de imágenes de referencia se identifican como RefPicList0 y RefPicList1. La construcción de la lista de imágenes de referencia incluye una etapa de inicialización de la lista de imágenes de referencia y una etapa de reordenación (modificación) de la lista de imágenes de referencia.

25 [0155] La inicialización de la lista de imágenes de referencia es un mecanismo explícito en el que el codificador de vídeo pone las imágenes de referencia almacenadas en la memoria de imágenes de referencia (también conocida como búfer de imágenes descodificadas) en una lista de imágenes de referencia en base al orden de los valores de POC (recuento de orden de imágenes, alineado con el orden de presentación de una imagen). Para 30 reordenar la lista de imágenes de referencia, el codificador de vídeo puede modificar la posición de una imagen en la lista durante la inicialización de la lista de imágenes de referencia a cualquier nueva posición, o poner cualquier imagen de referencia almacenada en la memoria de imágenes de referencia en cualquier posición. Esto incluye imágenes que no pertenecían a la lista inicializada. En algunos ejemplos, después de reordenar (modificar) la lista de imágenes de referencia, el codificador de vídeo coloca algunas imágenes en una posición más alejada de la lista. Sin embargo, si la posición de una imagen supera el número de imágenes de referencia activas de la lista, la imagen no se considera una entrada de la lista final de imágenes de referencia. El número de imágenes de referencia activas se puede señalizar en la cabecera de fragmento para cada lista. Despues de que se construyen 35 las listas de imágenes de referencia (a saber, RefPicList0 y RefPicList1 si están disponibles), el codificador de vídeo identifica una imagen de referencia incluida en la lista de imágenes de referencia en base a un índice de referencia en una lista de imágenes de referencia.

40 [0156] Como se describe anteriormente, un vector de movimiento identifica un bloque en una imagen de referencia. En algunos ejemplos, en lugar de codificar el vector de movimiento por sí mismo, el codificador de vídeo 45 puede utilizar predictores de vector de movimiento para derivar el vector de movimiento. Un ejemplo de un predictor de vector de movimiento es un predictor de vector de movimiento temporal. Para determinar un predictor de vector de movimiento temporal (TMVP), el codificador de vídeo identifica una imagen colocalizada. Si la imagen actual es un fragmento B, se señala un collocated\_from\_10\_flag (indicador colocalizado desde 10) en una cabecera de fragmento por el codificador de vídeo 20 para la recepción por el descodificador de vídeo 30 para indicar si la imagen colocalizada es de RefPicList0 o RefPicList1. Por ejemplo, un valor de 0 para el collocated\_from\_10\_flag indica que la imagen colocalizada se identifica en RefPicList1 y un valor de 1 para el collocated\_from\_10\_flag indica que la imagen colocalizada se identifica en RefPicList0.

50 [0157] Despues de que el codificador de vídeo identifica una lista de imágenes de referencia, el codificador de vídeo usa el collocated\_ref\_idx, señalado en una cabecera de fragmento, para identificar la imagen en la lista de imágenes de referencia identificadas. El codificador de vídeo identifica una unidad de predicción (PU) colocalizada en la imagen colocalizada. Los ejemplos de la unidad de predicción colocalizada incluyen una PU localizada en la parte inferior derecha con respecto a una CU en la imagen colocalizada en relación con donde está localizado el bloque actual en la imagen actual o un bloque central en la CU en la imagen colocalizada en relación con donde 55 está localizado el bloque actual. El TMVP puede ser una de la información de movimiento de la PU inferior derecha o la PU central.

[0158] En algunos ejemplos, el TMVP es uno de los predictores de vectores de movimiento que el codificador de vídeo identifica en una lista de predictores de vectores de movimiento candidatos usados para la predicción avanzada de vector de movimiento (AMVP) o el modo de fusión/salto, de los que ambos se describen con más detalle. Los ejemplos adicionales de predictores de vectores de movimiento incluyen vectores de movimiento de

5 bloques espacialmente vecinos denominados predictores de vectores de movimiento espacial (SMVP). De acuerdo con las técnicas descritas en la presente divulgación, el vector de movimiento del bloque de referencia identificado por el vector de disparidad también puede formar uno de los predictores de vectores de movimiento y, en algunos ejemplos, el vector de disparidad por sí mismo puede formar uno de los predictores de vectores de movimiento.

10 [0159] Cuando el codificador de vídeo usa vectores de movimiento identificados por el procedimiento anterior para generar un candidato de movimiento para AMVP o modo de fusión, el codificador de vídeo puede escalar los vectores de movimiento en base a la localización temporal (reflejada por POC). En algunos ejemplos, el codificador de vídeo se puede preconfigurar para establecer el índice de referencia para el TMVP igual a cero para el modo de fusión, y para el modo AMVP, el codificador de vídeo puede establecer el índice de referencia igual al índice de

15 referencia señalado.

[0160] En la HEVC, el SPS incluye un indicador `sps_temporal_mvp_enable_flag` y la cabecera de fragmento incluye un indicador `pic_temporal_mvp_enable_flag` cuando el `sps_temporal_mvp_enable_flag` es igual a 1. Cuando tanto el `pic_temporal_mvp_enable_flag` como el `id temporal` son iguales a 0 para una imagen particular, 20 ningún vector de movimiento de imágenes antes de esa imagen particular en el orden de descodificación se usaría como un predictor de vector de movimiento temporal en la descodificación de la imagen particular o de una imagen después de la imagen particular en el orden de descodificación.

[0161] Lo siguiente describe el modo de fusión y el modo AMVP. Las técnicas para usar un vector de disparidad como parte del modo de fusión y el modo AMVP se describen con más detalle a continuación adicionalmente.

[0162] En el modo de fusión y el modo AMVP, el codificador de vídeo (por ejemplo, el codificador de vídeo 20 o el descodificador de vídeo 30) construye una lista de predictores de vectores de movimiento candidatos. La lista de predictores de vectores de movimiento candidatos incluye vectores de movimiento para bloques vecinos, tales 30 como bloques vecinos espacial o temporalmente. El codificador de vídeo 20 señala un índice a la lista de predictores de vectores de movimiento candidatos, y el descodificador de vídeo 30 determina el predictor de vector de movimiento en base al índice.

[0163] En el modo de fusión, el codificador de vídeo hereda la totalidad de la información de movimiento para el predictor de vector de movimiento (por ejemplo, la imagen de referencia y el valor del vector de movimiento) como información de movimiento para el bloque actual. En el modo AMVP, el codificador de vídeo 20 señala una diferencia de vector de movimiento (MVD) entre el vector de movimiento real para el bloque actual y el predictor de vector de movimiento. El descodificador de vídeo 30 determina el vector de movimiento real para el bloque actual sumando la MVD con el predictor de vector de movimiento. Además, el codificador de vídeo 20 señala la información usada para identificar la(s) imagen/imágenes de referencia (por ejemplo, un valor de índice en una o ambas de `RefPicList0` y `RefPicList1`), y el descodificador de vídeo 30 determina la(s) imagen/imágenes de referencia en base a la información recibida usada para identificar la(s) imagen/imágenes de referencia.

[0164] En algunos ejemplos, además de los vectores de movimiento de los bloques vecinos, los vectores de movimiento determinados a partir del vector de disparidad del bloque actual, así como el vector de disparidad, se pueden añadir a la lista de predictores de vectores de movimiento candidatos para el modo de fusión y el modo AMVP. Las técnicas para determinar el vector de movimiento a partir del vector de disparidad y las técnicas para determinar el vector de disparidad se describen a continuación adicionalmente.

[0165] Lo siguiente describe técnicas para la derivación del vector de disparidad (por ejemplo, el procedimiento de derivación del vector de disparidad). Para derivar un vector de disparidad, se usa el procedimiento denominado técnica de derivación del vector de disparidad basada en bloques vecinos (NBDV) en la 3D-HTM actual por el codificador de vídeo 20 y el descodificador de vídeo 30. Por ejemplo, el codificador de vídeo utiliza vectores de movimiento de disparidad de bloques vecinos espaciales y temporales para derivar el vector de disparidad de acuerdo con la técnica de derivación NBDV. En la técnica de derivación NBDV, el codificador de vídeo comprueba los vectores de movimiento de los bloques vecinos espaciales o temporales en un orden de comprobación fijo. Después de que el codificador de vídeo identifica un vector de movimiento de disparidad o un vector de disparidad implícita (IDV) (el vector de disparidad implícita se describe con más detalle a continuación), el codificador de vídeo termina el procedimiento de comprobación y se devuelve el vector de movimiento de disparidad identificado, y el codificador de vídeo convierte el vector de movimiento de disparidad identificado en el vector de disparidad. El codificador de vídeo usa el vector de disparidad como predicción de movimiento entre vistas y predicción de residuos entre vistas.

[0166] En algunos ejemplos, puede ser posible que el codificador de vídeo no identifique un vector de movimiento de disparidad de un bloque vecino o un IDV después de comprobar todos los bloques vecinos predefinidos. En dichos ejemplos, si el codificador de vídeo no identifica un vector de movimiento de disparidad o un IDV, el

codificador de vídeo determina un vector de disparidad cero para el bloque actual. El codificador de vídeo usa un vector de disparidad cero para diversas herramientas de codificación de vídeo, tales como la predicción de movimiento entre vistas. Sin embargo, si el codificador de vídeo no identifica un vector de movimiento de disparidad o un IDV, la predicción residual entre vistas no está disponible para el bloque actual.

5 [0167] Como se describe anteriormente, el codificador de vídeo comprueba bloques vecinos espaciales y temporales para identificar un vector de movimiento de disparidad como parte de la técnica de derivación de NBDV. Para bloques vecinos espaciales, el codificador de vídeo comprueba cinco bloques vecinos espaciales, de los que los ejemplos incluyen los bloques de abajo a la izquierda, a la izquierda, arriba a la derecha, arriba y arriba a la izquierda de un bloque actual (por ejemplo, la unidad de predicción (PU) actual) indicados por A0, A1, B0, B1 o B2, como se define en la figura 8-3 de la especificación HEVC. Los bloques vecinos temporales que comprueba el codificador de vídeo se describen con más detalle con respecto a la FIG. 5.

10 [0168] La FIG. 5 es un diagrama conceptual que ilustra bloques vecinos temporales para la derivación del vector de disparidad basado en bloques vecinos (NBDV). Para los bloques vecinos temporales, el codificador de vídeo comprueba hasta dos imágenes de referencia de una vista actual. Las dos imágenes de referencia son la imagen colocalizada (por ejemplo, la que se señala en la cabecera de fragmento como parte del procedimiento TMVP descrito anteriormente) y una de la imagen de acceso aleatorio o la imagen de referencia con la diferencia de POC más pequeña y el ID temporal más pequeño. En algunos ejemplos, el codificador de vídeo, en primer lugar comprueba la imagen de acceso aleatorio, seguido de la imagen colocalizada. Por ejemplo, si el codificador de vídeo identifica un vector de movimiento de disparidad en la imagen de acceso aleatorio, el codificador de vídeo termina la comprobación de bloques vecinos temporales. Si el codificador de vídeo no identifica un vector de movimiento de disparidad en la imagen de acceso aleatorio, el codificador de vídeo comprueba la imagen colocalizada (por ejemplo, la imagen identificada en la cabecera de fragmento como parte del procedimiento TMVP). Cada una de estas imágenes se denomina imagen candidata para los propósitos de la derivación NBDV.

15 [0169] Para cada imagen candidata, el codificador de vídeo comprueba dos bloques candidatos. El primer bloque es el bloque central (CR). Por ejemplo, el bloque CR está localizado en la pos. A en la FIG. 5, y es el bloque 4x4 central de la región colocalizada de la PU actual. El segundo bloque es el bloque inferior derecho (BR). Por ejemplo, el bloque BR está localizado en la pos. B en la FIG. 5, y es el bloque 4x4 inferior derecho de la región colocalizada de la PU actual.

20 [0170] Como se describe anteriormente, además de comprobar bloques vecinos espaciales y temporales para un vector de movimiento de disparidad, el codificador de vídeo puede identificar un vector de disparidad implícita (IDV). Se genera un vector de disparidad implícita (IDV) cuando una PU emplea predicción de vector de movimiento entre vistas, es decir, el candidato para los modos AMVP o de fusión se deriva de un bloque correspondiente en la otra vista con la ayuda de un vector de disparidad. Un vector de disparidad de este tipo se denomina IDV. Un IDV se almacena en la PU con el propósito de la derivación del vector de disparidad. En otras palabras, el codificador de vídeo puede haber implementado técnicas de NBDV para derivar un vector de disparidad para un bloque vecino. El vector de disparidad para el bloque vecino puede funcionar como un vector de disparidad para el bloque actual. El vector de disparidad para el bloque vecino que funciona como un vector de disparidad para el bloque actual se denomina IDV.

25 [0171] El codificador de vídeo puede comprobar los bloques vecinos espaciales, los bloques vecinos temporales y el IDV en un orden de comprobación específico. Por ejemplo, el codificador de vídeo puede comprobar, en primer lugar, los bloques vecinos espaciales y temporales para determinar un vector de movimiento de disparidad, y si no se identifica ningún vector de movimiento de disparidad, puede comprobar si hay un IDV.

30 [0172] En algunos ejemplos, el codificador de vídeo, en primer lugar comprueba los bloques vecinos espaciales y a continuación los bloques vecinos temporales. Por ejemplo, el codificador de vídeo comprueba los cinco bloques vecinos espaciales en el siguiente orden: A1, B1, B0, A0 y B2. Si uno de ellos usa un vector de movimiento de disparidad (DMV), el codificador de vídeo termina el procedimiento de comprobación y usa el DMV correspondiente como vector de disparidad final. Para el bloque vecino temporal, para cada imagen candidata, el codificador de vídeo comprueba los dos bloques en el siguiente orden: CR (pos. A en la FIG. 5) y BR (pos. B en la FIG. 5) para la primera vista no básica (por ejemplo, imagen de acceso aleatorio) o BR, CR para la segunda vista no básica (por ejemplo, para la imagen colocalizada). Si uno de ellos usa DMV, el codificador de vídeo termina el procedimiento de comprobación y usa el DMV correspondiente como vector de disparidad final. Para IDV, el codificador de vídeo comprueba los cinco bloques vecinos espaciales en el siguiente orden: A0, A1, B0, B1 y B2. Si uno de ellos usa IDV (es decir, si existe un vector de disparidad para uno de ellos), y el bloque se codifica como modo de salto/fusión, el codificador de vídeo termina, el procedimiento de comprobación se termina y usa el IDV correspondiente como vector de disparidad final.

35 [0173] En algunos ejemplos, el codificador de vídeo puede refinar adicionalmente el vector de disparidad. Por ejemplo, el codificador de vídeo usa información en el mapa de profundidad codificado para refinar el vector de disparidad generado a partir del esquema NBDV. Por ejemplo, la exactitud del vector de disparidad derivado simplemente de la técnica de derivación de NBDV puede no ser lo suficientemente exacta, y la exactitud se puede

incrementar usando información en el mapa de profundidad de vista básica codificada. Por ejemplo, en la FIG. 3, la vista SO es la vista básica, y para cada uno de los componentes de vista de textura de la vista S0, puede haber un mapa de profundidad correspondiente. El codificador de vídeo puede realizar un refinamiento del vector de disparidad usando el mapa de profundidad de los componentes de vista de textura correspondientes en la vista básica S0.

5 [0174] Por ejemplo, el codificador de vídeo puede localizar un bloque de profundidad correspondiente por el vector de disparidad derivado de NBDV en la vista de profundidad de referencia codificada previamente, tal como la vista básica. El tamaño del bloque de profundidad correspondiente es el mismo que el de la PU actual. El codificador de vídeo puede calcular un vector de disparidad a partir del bloque de profundidad colocalizado en base al valor máximo de los valores de profundidad de las cuatro esquinas. El codificador de vídeo establece el valor de profundidad máximo igual a la componente horizontal de un vector de disparidad, y establece la componente vertical del vector de disparidad en 0.

10 [0175] El vector de disparidad refinado resultante se denomina "vector de disparidad basado en bloques vecinos orientados en profundidad" (DoNBDV). El codificador de vídeo reemplaza el vector de disparidad del esquema NBDV con el vector de disparidad recién derivado del esquema DoNBDV para la derivación del candidato entre vistas para los modos AMVP y de fusión. Sin embargo, para la predicción residual entre vistas, el codificador de vídeo usa el vector de disparidad sin refinar. Además, el vector de disparidad refinado se almacena como el vector de movimiento de una PU si está codificado con el modo de predicción de síntesis de vista hacia atrás (VSP). En la presente divulgación, el término vector de disparidad se puede referir a un vector de disparidad sin refinar o un vector de disparidad refinado.

15 [0176] La FIG. 6 es un diagrama conceptual que ilustra la derivación de bloques de profundidad a partir de una vista de referencia asociada con la predicción de síntesis de vista hacia atrás (BVSP). Por ejemplo, la FIG. 6 es un diagrama conceptual que ilustra técnicas relacionadas con BVSP usando bloques vecinos. Se ha propuesto y adoptado BVSP como técnica para 3D-HEVC. El enfoque de VSP de distorsión inversa como se propone en JCT3V-C0152 se adoptó en la tercera conferencia del JCT-3V. JCT3V-C0152 está disponible en [http://phenix.int-evry.fr/jct3v/doc\\_end\\_user/current\\_document.php?id=594](http://phenix.int-evry.fr/jct3v/doc_end_user/current_document.php?id=594).

20 [0177] La idea básica de esta VSP de distorsión inversa es la misma que la VSP basada en bloques en 3D-AVC. Ambas técnicas usan la distorsión inversa y la VSP basada en bloques para evitar transmitir las diferencias de vectores de movimiento y usar vectores de movimiento más precisos. Los detalles de implementación son diferentes debido a las diferentes plataformas. En los siguientes párrafos, el término "BVSP" se usa para indicar el enfoque de VSP de distorsión inversa en 3D-HEVC.

25 [0178] En 3D-HTM, la codificación de textura primero se aplica en condiciones de prueba comunes. Por lo tanto, la vista de profundidad no básica correspondiente no está disponible cuando se descodifica una vista de textura no básica. Por lo tanto, la información de profundidad se estima y se usa para realizar la BVSP. Para estimar la información de profundidad para un bloque, algunas técnicas en primer lugar derivan un vector de disparidad de los bloques vecinos, y a continuación usan el vector de disparidad derivado para obtener un bloque de profundidad a partir de una vista de referencia.

30 [0179] En el modelo de prueba HTM 5.1, existe un procedimiento para derivar un predictor de vector de disparidad, conocido como NBDV. Si se deja que  $(dvx, dvy)$  indique el vector de disparidad identificado a partir de la función NBDV, y la posición del bloque actual es  $(blockx, blocky)$ . Se propuso buscar un bloque de profundidad en  $(blockx+dvx, blocky+dvy)$  en la imagen de profundidad de la vista de referencia. El bloque de profundidad buscado tendría el mismo tamaño que la unidad de predicción (PU) actual, y se usaría a continuación para realizar una distorsión inversa para la PU actual. La FIG. 6 ilustra las etapas para localizar un bloque de profundidad desde la vista de referencia y a continuación usar el bloque de profundidad para la predicción de BVSP.

35 [0180] En el ejemplo de la FIG. 6, la imagen de profundidad 150 y la imagen de textura 154 corresponden a la misma vista, mientras que la imagen de textura 152 corresponde a una vista diferente. En particular, la imagen de textura 152 incluye el bloque actual 160 que está codificado en relación con la imagen de textura 154, actuando como imagen de referencia. Un codificador de vídeo se puede referir al bloque vecino 162, que es vecino al bloque actual 160. El bloque vecino 162 incluye un vector de disparidad 166 determinado previamente. El vector de disparidad 166 se puede derivar como un vector de disparidad 164 para el bloque actual 160. Por tanto, el vector de disparidad 164 se refiere al bloque de profundidad 156 en la imagen de profundidad 150 de la vista de referencia.

40 [0181] A continuación, el codificador de vídeo puede usar píxeles (es decir, valores de profundidad) del bloque de profundidad 156 para determinar los valores de disparidad 168 para píxeles (es decir, valores de textura) del bloque actual 160, para realizar distorsión inversa. A continuación, el codificador de vídeo puede sintetizar valores para un bloque predicho (es decir, un bloque de referencia BVSP) para el bloque actual 160 a partir de los píxeles identificados por los valores de disparidad 168. A continuación, el codificador de vídeo puede predecir el bloque actual 160 usando este bloque predicho. Por ejemplo, durante la codificación de vídeo por el codificador de vídeo 20, el codificador de vídeo 20 puede calcular las diferencias píxel por píxel entre el bloque predicho y el bloque

actual 160 para producir un valor residual, que el codificador de vídeo 20 a continuación puede transformar, cuantificar y codificar por entropía. Por otra parte, durante la descodificación de vídeo por el descodificador de vídeo 30, el descodificador de vídeo 30 puede descodificar por entropía, cuantificar inversamente y transformar inversamente los datos residuales, a continuación combinar los datos residuales (en base de pixel por pixel) con el bloque predicho para reproducir el bloque actual 160.

**[0182]** En algunos ejemplos, si se habilita la BVSP en la secuencia, se cambia el procedimiento de NBDV por la predicción de movimiento entre vistas. Para cada uno de los bloques vecinos temporales, si usa un vector de movimiento de disparidad, el codificador de vídeo devuelve el vector de movimiento de disparidad como vector de disparidad y refina adicionalmente el vector de disparidad con el procedimiento descrito anteriormente con respecto a las técnicas para refinar adicionalmente el vector de disparidad. Para cada uno de los bloques vecinos espaciales, se aplica lo siguiente. Para cada lista de imágenes de referencia 0 o lista de imágenes de referencia 1, se aplica lo siguiente, si usa un vector de movimiento de disparidad, el codificador de vídeo devuelve el vector de movimiento de disparidad como vector de disparidad y se refina adicionalmente con el procedimiento descrito anteriormente con respecto a las técnicas para refinar adicionalmente el vector de disparidad. De otro modo, si usa el modo BVSP, el codificador de vídeo devuelve el vector de movimiento asociado como vector de disparidad. El vector de disparidad se refina adicionalmente de un modo similar al descrito anteriormente con respecto a las técnicas para refinar adicionalmente el vector de disparidad. Sin embargo, el valor máximo de profundidad se selecciona a partir de todos los píxeles del bloque de profundidad correspondiente, en lugar de cuatro píxeles de esquina. Para cada uno de los bloques vecinos espaciales, si usa un IDV, el IDV se devuelve como vector de disparidad y se refina adicionalmente con el procedimiento descrito anteriormente con respecto a las técnicas para refinar adicionalmente el vector de disparidad. Como se describe anteriormente, en la presente divulgación, el término "vector de disparidad" se usa genéricamente para referirse a un vector de disparidad sin refinar o un vector de disparidad refinado.

**[0183]** Para la indicación de PU codificadas con BVSP, el codificador de vídeo trata el modo BVSP introducido como un modo intercodificado especial y usa un indicador para indicar el uso del modo BVSP, que el codificador de vídeo puede mantener para cada PU. En algunos ejemplos, en lugar de señalar el indicador en el flujo de bits, el codificador de vídeo añade un nuevo candidato de fusión (candidato de fusión BVSP) a la lista de candidatos de fusión (por ejemplo, lista de predictores de vectores de movimiento candidatos), y el indicador depende de si el índice de candidatos de fusión descodificados corresponde a un candidato de fusión BVSP.

**[0184]** Lo siguiente describe el candidato de fusión BVSP. Por ejemplo, el codificador de vídeo puede establecer el índice de imagen de referencia para cada lista de imágenes de referencia en -1, y a continuación establecer el vector de movimiento para cada lista de imágenes de referencia como vector de disparidad refinado.

**[0185]** La posición insertada del candidato de fusión BVSP depende de los bloques vecinos espaciales. Por ejemplo, si cualquiera de los cinco bloques vecinos espaciales (A0, A1, B0, B1 o B2) se codifica con el modo BVSP (es decir, el indicador mantenido del bloque vecino es igual a 1), el codificador de vídeo trata el candidato de fusión BVSP como el candidato de fusión espacial correspondiente y lo inserta en la lista de candidatos de fusión. En algunos ejemplos, el codificador de vídeo inserta el candidato de fusión BVSP solo una vez en la lista de candidatos de fusión. De otro modo (por ejemplo, si ninguno de los cinco bloques vecinos espaciales se codifica con el modo BVSP), el codificador de vídeo inserta 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 procedimiento de derivación de candidatos de fusión bipredictivo combinado, el codificador de vídeo puede comprobar condiciones adicionales para evitar incluir el candidato de fusión BVSP.

**[0186]** Lo siguiente describe el procedimiento de derivación de predicciones. Para cada PU codificada con BVSP con su tamaño indicado por  $N \times M$ , el codificador de vídeo divide además la PU en varias subregiones con el tamaño igual a  $K \times K$  (donde  $K$  puede ser 4 o 2). Para cada subregión, el codificador de vídeo deriva un vector de movimiento de disparidad separado y predice cada subregión a partir de un bloque localizado por el vector de movimiento de disparidad derivado en la imagen de referencia entre vistas. En otras palabras, el tamaño de las unidades de compensación de movimiento para las PU codificadas con BVSP se establece en  $K \times K$ . En algunas condiciones de prueba comunes,  $K$  se establece en 4.

**[0187]** Lo siguiente describe el procedimiento de derivación del vector de movimiento de disparidad. Para cada subregión (por ejemplo, un bloque 4x4) dentro de una PU codificada con el modo BVSP, el codificador de vídeo en primer lugar localiza un bloque de profundidad 4x4 correspondiente en la vista de profundidad de referencia con el vector de disparidad refinado descrito anteriormente. A continuación, el codificador de vídeo selecciona el valor máximo de los dieciséis píxeles de profundidad en el bloque de profundidad correspondiente. Seguidamente, el codificador de vídeo convierte el valor máximo en la componente horizontal de un vector de movimiento de disparidad. La componente vertical del vector de movimiento de disparidad se establece en 0.

**[0188]** Como se describe anteriormente, una de las herramientas de codificación de vídeo que usa el vector de disparidad es la predicción de movimiento entre vistas. Lo siguiente describe la predicción de movimiento entre vistas, incluyendo la manera en que se usa el vector de disparidad.

5 [0189] La FIG. 7 es un diagrama de bloques que ilustra la derivación del candidato de vector de movimiento predicho entre vistas. En base al vector de disparidad derivado del esquema DoNBDV, se puede añadir un nuevo candidato de vector de movimiento, el candidato de vector de movimiento predicho entre vistas (IPMVC), si está disponible, a AMVP y los modos de salto/fusión por el codificador de vídeo. El vector de movimiento predicho entre vistas, si está disponible, es un vector de movimiento temporal.

10 10 [0190] Debido a que el modo de salto tiene el mismo procedimiento de derivación del vector de movimiento que el modo de fusión, las técnicas descritas a continuación se pueden aplicar tanto a los modos de fusión como a los de salto. Para el modo de fusión/salto, el codificador de vídeo deriva el vector de movimiento predicho entre vistas de acuerdo con las siguientes etapas. El codificador de vídeo localiza un bloque correspondiente de un bloque actual (por ejemplo, la PU/CU actual) en una vista de referencia de la misma unidad de acceso usando el vector de disparidad. Si el bloque correspondiente no está intracodificado ni predicho entre vistas, y su imagen de referencia tiene un valor de POC igual al de una entrada en la misma lista de imágenes de referencia de la PU/CU actual, el codificador de vídeo convierte el índice de referencia en base a el valor de POC, y deriva la información de movimiento (dirección de predicción, imágenes de referencia y vectores de movimiento) del bloque correspondiente para que sea el vector de movimiento predicho entre vistas.

15 20 [0191] Por ejemplo, en la FIG. 7, el bloque actual está en la vista V1 en el tiempo T1. El codificador de vídeo puede identificar el bloque correspondiente para el bloque actual a partir de la imagen de referencia en el tiempo T1 en la vista V0. El codificador de vídeo puede derivar el vector predicho entre vistas como vector(es) de movimiento para el bloque correspondiente en la imagen de referencia en la vista V0, como se ilustra por las flechas verticales discontinuas.

25 [0192] En algunos ejemplos, el bloque correspondiente se identifica usando las siguientes ecuaciones para determinar una localización en la imagen de referencia en una vista de referencia, y a continuación el bloque que cubre la localización determinada se identifica como el bloque de referencia (por ejemplo, el bloque correspondiente).

$$xRef = Clip3( 0, PicWidthInSamples_L - 1, xP + ( ( nPSW - 1 ) >> 1 ) + ( ($$

$$30 mvDisp[ 0 ] + 2 ) >> 2 ) ) \quad (H-124)$$

$$yRef = Clip3( 0, PicHeightInSamples_L - 1, yP + ( ( nPSH - 1 ) >> 1 ) + ( ($$

$$50 mvDisp[ 1 ] + 2 ) >> 2 ) ) \quad (H-125)$$

35 [0193] En otras palabras, en 3D-HEVC, el bloque 4x4 de referencia se identifica por dos etapas, la primera etapa es identificar un píxel con un vector de disparidad y la segunda etapa es obtener el bloque 4x4 (con un conjunto único de información de movimiento correspondiente a RefPicList0 o RefPicList1, respectivamente) y utilizando la información de movimiento para crear un candidato de fusión. En algunos ejemplos, la ecuación anterior se usa para determinar la localización en la imagen de referencia de la vista de referencia (por ejemplo, para determinar la localización de la localización C0 de la FIG. 4).

40 40 [0194] Sin embargo, de acuerdo con las técnicas descritas en la presente divulgación, el uso de la ecuación anterior para determinar una localización en la imagen de referencia puede no tener estadísticamente a proporcionar las ganancias óptimas de codificación de vídeo. En consecuencia, en algunos ejemplos, se pueden usar las siguientes ecuaciones para determinar la localización dentro de la imagen de referencia (por ejemplo, determinar la localización de la localización C3 de la FIG. 4).

$$50 xRef = Clip3( 0, PicWidthInSamples_L - 1, xP + ( ( nPSW ) >> 1 ) + ( ($$

$$mvDisp[ 0 ] + 2 ) >> 2 ) ) (H-124)$$

$$yRef = Clip3( 0, PicHeightInSamples_L - 1, yP + ( ( nPSH ) >> 1 ) + ( ($$

$$55 mvDisp[ 1 ] + 2 ) >> 2 ) ) (H-125)$$

50 [0195] Además, en algunos ejemplos, el codificador de vídeo convierte el vector de disparidad en un vector de movimiento de disparidad entre vistas y añade el vector de movimiento de disparidad entre vistas a la lista de candidatos de fusión en una posición diferente del IPMVC, o añade el vector de movimiento de disparidad entre vistas a la lista de candidatos de AMVP en la misma posición que IPMVC cuando está disponible. En este contexto, el IPMVC o el candidato de vector de movimiento de disparidad entre vistas (IDMVC) se denomina "candidato entre vistas".

55 [0196] En el modo de fusión/salto, el IPMVC, si está disponible, el codificador de vídeo inserta el IPMVC antes

de todos los candidatos de fusión espacial y temporal a la lista de candidatos de fusión. El codificador de vídeo inserta el IDMVC antes del candidato de fusión espacial derivado de A0.

5 [0197] Lo siguiente describe la construcción de la lista de candidatos de fusión para codificación de textura en 3D-HEVC. El codificador de vídeo deriva el vector de disparidad mediante el procedimiento de DoNBDV. Con el vector de disparidad, el procedimiento de construcción de la lista de candidatos de fusión en 3D-HEVC se define como sigue.

10 [0198] El codificador de vídeo deriva el IPMVC por el procedimiento descrito anteriormente. Si está disponible, el codificador de vídeo inserta el IPMVC en la lista de fusión. A continuación, en 3D-HEVC, el codificador de vídeo realiza el procedimiento de derivación para los candidatos de fusión espacial y la inserción de IDMVC. Por ejemplo, el codificador de vídeo puede comprobar la información de movimiento de las PU vecinas espaciales en el siguiente orden: A1, B1, B0, A0 o B2.

15 [0199] El codificador de vídeo puede realizar una depuración restringida por los siguientes procedimientos. Si A1 e IPMVC tienen los mismos vectores de movimiento y los mismos índices de referencia, el codificador de vídeo no inserta A1 en la lista de candidatos. De otro modo, el codificador de vídeo inserta A1 en la lista. Si B1 y A1/IPMVC tienen los mismos vectores de movimiento y los mismos índices de referencia, el codificador de vídeo no inserta B1 en la lista de candidatos. De otro modo, el codificador de vídeo inserta B1 en la lista. Si B0 está disponible, el codificador de vídeo añade B0 a la lista de candidatos. El IDMVC se deriva por el procedimiento descrito anteriormente. Si el IDMVC está disponible y es diferente de los candidatos derivados de A1 y B1, el codificador de vídeo inserta el IDMVC en la lista de candidatos. Si se habilita BVSP para la imagen completa o para el fragmento actual, a continuación el codificador de vídeo inserta el candidato de fusión BVSP en la lista de candidatos de fusión. Si A0 está disponible, el codificador de vídeo añade A0 a la lista de candidatos. Si B2 está disponible, el codificador de vídeo lo añade a la lista de candidatos.

20 [0200] Seguidamente, el codificador de vídeo realiza el procedimiento de derivación del candidato de fusión temporal. De manera similar al procedimiento de derivación de candidatos de fusión temporal en HEVC, cuando se utiliza la información de movimiento de la PU colocalizada, el índice de imagen de referencia objetivo del candidato de fusión temporal se puede cambiar, en lugar de fijarlo para que sea 0. Cuando el índice de referencia objetivo igual a 0 corresponde a una imagen de referencia temporal (en la misma vista) mientras que el vector de movimiento de la unidad de predicción (PU) colocalizada apunta a una imagen de referencia entre vistas, el codificador de vídeo cambia el índice de referencia a otro índice que corresponde a la primera entrada de una imagen de referencia entre vistas en la lista de imágenes de referencia (por ejemplo, la primera entrada con una imagen de referencia en otra vista). Cuando el índice de referencia objetivo igual a 0 corresponde a una imagen de referencia entre vistas mientras que el vector de movimiento de la unidad de predicción (PU) colocalizada apunta a una imagen de referencia temporal, el codificador de vídeo cambia el índice de referencia a otro índice que corresponde a la primera entrada de la imagen de referencia temporal en la lista de imágenes de referencia (por ejemplo, la primera entrada con una imagen de referencia en la misma vista).

30 [0201] A continuación sigue el procedimiento de derivación para candidatos de fusión bipredictiva combinados en 3D-HEVC. Si el número total de candidatos derivados de las dos etapas anteriores es menor que el número máximo de candidatos, el codificador de vídeo realiza el mismo procedimiento definido en HEVC, excepto la especificación de 10CandIdx e I1CandIdx. La relación entre combIdx, 10CandIdx e I1CandIdx se define en la siguiente tabla:

40 **Tabla 1 - Especificación de I0CandIdx e I1CandIdx en 3D-HEVC**

comIdx	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
I0CandIdx	0	1	0	2	1	2	0	3	1	3	2	3	0	4	1	4	2	4	3	4
I1CandIdx	1	0	2	0	2	1	3	0	3	1	3	2	4	0	4	1	4	2	4	3

50 [0202] A continuación sigue el procedimiento de derivación para los candidatos de fusión de vector de movimiento cero. El codificador de vídeo puede aplicar el mismo procedimiento como se define en HEVC para los candidatos de fusión de vector de movimiento cero.

55 [0203] En el software más reciente, el número total de candidatos en la lista de fusión (MRG) es de hasta 6 y five\_minus\_max\_num\_merge\_cand (cinco menos el número máximo de candidatos de fusión) se señala para especificar el número máximo de candidatos MRG restados de 6 en la cabecera de fragmento, el elemento de sintaxis five\_minus\_max\_num\_merge\_cand está en el intervalo de 0 a 5, inclusive.

60 [0204] La FIG. 8 es un diagrama conceptual que ilustra la derivación de un candidato de vector de movimiento heredado (MVI) para codificación de profundidad. Lo siguiente describe el vector de movimiento heredado para la

codificación de profundidad. Una idea detrás del vector de movimiento heredado (MVI) es aprovechar la similitud de las características de movimiento entre las imágenes de textura y las imágenes de profundidad asociadas.

5 **[0205]** Para una PU dada en la imagen de profundidad, el candidato MVI reutiliza los vectores de movimiento y los índices de referencia del bloque de textura correspondiente ya codificado, si está disponible. Es decir, la información de movimiento para un bloque de textura se usa como información de movimiento para el bloque de profundidad correspondiente para la misma imagen y vista. La FIG. 8 muestra un ejemplo del procedimiento de derivación del candidato MVI donde se selecciona el bloque de textura correspondiente como bloque 4 x 4 localizado a la derecha y en la parte inferior del centro de la PU actual.

10 **[0206]** En algunos ejemplos, el codificador de vídeo usa vectores de movimiento con precisión de entero en la codificación de profundidad, y usa un cuarto de precisión de los vectores de movimiento para la codificación de textura. Por lo tanto, el codificador de vídeo puede (por ejemplo, deberá) escalar el vector de movimiento del bloque de textura correspondiente antes de usarlo como candidato MVI.

15 **[0207]** Con la generación de candidatos MVI, la lista de candidatos de fusión para las vistas de profundidad se construye como sigue. El codificador de vídeo deriva el MVI en base al procedimiento descrito anteriormente. Si está disponible, el codificador de vídeo inserta el MVI en la lista de fusión. Seguidamente, el codificador de vídeo puede realizar el procedimiento de derivación para los candidatos de fusión espacial y la inserción de IDMVC en 20 3D-HEVC. Por ejemplo, el vídeo puede comprobar la información de movimiento de las PU vecinas espaciales en el siguiente orden: A1, B1, B0, A0 o B2.

25 **[0208]** El codificador de vídeo puede realizar una depuración restringida usando los siguientes procedimientos. Si A1 y MVI tienen los mismos vectores de movimiento y los mismos índices de referencia, el codificador de vídeo no inserta A1 en la lista de candidatos. Si B1 y A1/MVI tienen los mismos vectores de movimiento y los mismos índices de referencia, el codificador de vídeo no inserta B1 en la lista de candidatos. Si B0 está disponible, el codificador de vídeo añade B0 a la lista de candidatos. Si A0 está disponible, el codificador de vídeo añade A0 a la lista de candidatos. Si B2 está disponible, el codificador de vídeo añade B2 a la lista de candidatos.

30 **[0209]** A continuación, el codificador de vídeo puede realizar el procedimiento de derivación para el candidato de fusión temporal. Por ejemplo, el codificador de vídeo puede realizar etapas similares a las del procedimiento de derivación de candidatos de fusión temporal en HEVC, donde se utiliza la información de movimiento de la PU colocalizada. Sin embargo, el índice de imágenes de referencia objetivo del candidato de fusión temporal se puede cambiar como se explica anteriormente con respecto a la descripción de la construcción de la lista de candidatos 35 de fusión en lugar de fijarlo para que sea 0.

40 **[0210]** A continuación, el codificador de vídeo puede implementar el procedimiento de derivación para candidatos de fusión bipredictiva combinados en 3D-HEVC. Por ejemplo, si el número total de candidatos derivados de las dos etapas anteriores es menor que el número máximo de candidatos, el codificador de vídeo puede realizar el mismo procedimiento definido en HEVC, excepto la especificación de I0CandIdx e I1CandIdx. La relación entre 45 combIdx, I0CandIdx e I1CandIdx se define en la tabla 1 (anteriormente) de la presente divulgación.

**[0211]** A continuación, el codificador de vídeo puede implementar el procedimiento de derivación para candidatos de fusión de vector de movimiento cero. Por ejemplo, el codificador de vídeo puede realizar el mismo procedimiento como se define en HEVC.

50 **[0212]** Como se describe anteriormente, además de la predicción de movimiento entre vistas, otra herramienta de codificación de vídeo que se basa en el vector de disparidad es la predicción residual entre vistas. Lo siguiente proporciona una descripción adicional para la predicción residual entre vistas.

55 **[0213]** La FIG. 9 es un diagrama conceptual que ilustra una estructura de predicción de ejemplo de predicción residual avanzada. En la 3D-HEVC actual, para utilizar más eficazmente la correlación entre la señal residual de dos vistas, la predicción residual entre vistas se realizó por la denominada predicción residual avanzada (ARP), en la que el residuo del bloque de referencia identificado con el vector de disparidad se generó sobre la marcha, como se representa en la FIG. 9, en lugar de mantener una imagen residual para la vista de referencia y predecir directamente el residuo dentro del bloque de referencia en la imagen residual.

60 **[0214]** Como se muestra en la FIG. 9, para predecir mejor el residuo del bloque actual en una vista no básica, indicado como Dc, el codificador de vídeo identifica en primer lugar el bloque de referencia Bc por el vector de disparidad, y se invoca la compensación de movimiento del bloque de referencia para derivar el residuo entre la señal de predicción Br y la señal reconstruida del bloque de referencia Bc. Cuando se invoca el modo ARP, el codificador de vídeo añade el residuo predicho por encima de la señal de predicción de la vista no básica, generada por la compensación de movimiento del bloque Dr en la imagen de referencia de la vista no básica. Una ventaja potencial del modo ARP es que el vector de movimiento usado por el bloque de referencia (cuando genera el residuo para ARP) está alineado con el vector de movimiento del bloque actual, por lo que la señal residual del bloque actual se puede predecir con mayor precisión. Por lo tanto, la energía del residuo se puede reducir 65

significativamente.

**[0215]** La FIG. 10 es otro diagrama conceptual que ilustra una estructura de predicción de ejemplo de predicción residual avanzada. Por ejemplo, la FIG. 10 ilustra diversos componentes de la FIG. 9, pero sin detalles de textura de las diferentes imágenes. Se apreciará que, sólo con propósitos de facilidad de ilustración, la FIG. 10 no está dibujada a escala con respecto a la FIG. 9.

**[0216]** Puesto que la(s) diferencia(s) de cuantificación entre las vistas básica (referencia) y no básica puede(n) dar lugar a una menor exactitud de predicción, el codificador de vídeo aplica de forma adaptativa dos factores de ponderación al residuo generado a partir de la vista de referencia: 0,5 y 1. Puesto que la compensación de movimiento adicional en la vista básica (referencia) puede requerir un incremento significativo en el acceso a la memoria y los cálculos, se han adoptado varios modos de hacer que el diseño sea más práctico con un sacrificio menor de la eficacia de codificación.

**[0217]** Como un ejemplo, el modo ARP solo se habilita cuando la unidad de predicción (PU) se codifica con  $2N \times 2N$ , para reducir los cálculos, especialmente por el codificador de vídeo 20. Además, se adoptan filtros bilineales para la compensación de movimiento tanto del bloque de referencia como del bloque actual, para reducir significativamente el acceso a la memoria para los bloques codificados con el modo ARP. Además, para mejorar la eficacia de la memoria caché, aunque los vectores de movimiento pueden apuntar a diferentes imágenes en la vista no básica, la imagen de referencia en la vista básica está fija. En este caso, es posible que sea necesario escalar el vector de movimiento del bloque actual en base a las distancias de la imagen.

**[0218]** Lo siguiente describe algunas otras técnicas para la intercodificación de profundidad. Por ejemplo, en las solicitudes provisionales de EE. UU. n.ºs 61/840.400, 61/847.942 y 61/890.107, presentadas el 27 de junio de 2013, el 18 de julio de 2013 y el 11 de octubre de 2013, respectivamente, y las solicitudes de patente de EE. UU. n.ºs 14/316.088 y 14/316.145, presentadas ambas el 26 de junio de 2014, se describen técnicas en las que, cuando se codifica una imagen de profundidad, un vector de disparidad se convierte en un valor de profundidad estimado de las muestras vecinas del bloque actual. Además, se pueden derivar más candidatos de fusión (por ejemplo, accediendo al bloque de referencia de la vista básica identificada por un vector de disparidad).

**[0219]** La FIG. 11 es un diagrama de bloques que ilustra un codificador de vídeo 20 de ejemplo que puede implementar o utilizar de otro modo técnicas para la identificación de bloques de acuerdo con una o más técnicas de ejemplo descritas en la presente divulgación. El codificador 20 de vídeo puede realizar intra e intercodificación (es decir, codificación por intrapredicción o codificación por interpredicción) de bloques de vídeo dentro de fragmentos de vídeo. La intracodificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial de un vídeo dentro de una trama o imagen de vídeo dada. La intercodificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal de un vídeo dentro de tramas o imágenes contiguas de una secuencia de vídeo. Intramodo (modo I) se puede referir a cualquiera de varios modos de codificación de base espacial. Los intermodos, tales como la predicción unidireccional (modo P) o la predicción bidireccional (modo B), se pueden referir a cualquiera de varios modos de codificación de base temporal.

**[0220]** Además, el codificador de vídeo 20 se puede configurar para realizar predicción entre vistas en la que una imagen de referencia reside en una vista distinta de la vista que incluye la imagen actual. En este sentido, la predicción entre vistas se puede considerar como una forma de intercodificación (por ejemplo, codificación por interpredicción). Además, el codificador de vídeo 20 se puede configurar para implementar herramientas de codificación de vídeo que se basan en un vector de disparidad como parte de la predicción entre vistas. Por ejemplo, el codificador de vídeo 20 se puede configurar para implementar la predicción de movimiento entre vistas y la predicción residual entre vistas, como dos ejemplos.

**[0221]** Para realizar la predicción de movimiento entre vistas y la predicción residual entre vistas, el codificador de vídeo 20 puede derivar un vector de disparidad para un bloque actual. En las técnicas descritas en la presente divulgación, el codificador de vídeo 20 puede establecer la posición de inicio del vector de disparidad para comenzar desde el píxel inferior derecho en el subbloque 2x2 central dentro del bloque actual, en lugar del píxel superior izquierdo en el subbloque 2x2 central. El codificador de vídeo 20 puede determinar el bloque al que se refiere el vector de disparidad en base a la posición de inicio establecida, y utilizar la información de movimiento del bloque para herramientas de codificación de vídeo que se basan en el vector de disparidad.

**[0222]** Como se muestra en la FIG. 11, el codificador de vídeo 20 recibe un bloque de vídeo actual dentro de una trama de vídeo que se va a codificar. En el ejemplo de la FIG. 10, el codificador de vídeo 20 incluye datos de memoria de vídeo 1100, una unidad de selección de modo 1140, una memoria de imágenes de referencia 1164, un sumador 1150, una unidad de procesamiento de transformada 1152, una unidad de cuantificación 1154 y una unidad de codificación por entropía 1156. A su vez, la unidad de selección de modo 1140 incluye una unidad de compensación de movimiento 1144, una unidad de estimación de movimiento 1142, una unidad de intrapredicción 1146 y una unidad de división 1148. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 también incluye una unidad de cuantificación inversa 1158, una unidad de procesamiento de transformada inversa 1160 y un sumador 1162. También se puede incluir un filtro de eliminación de bloques (no mostrado en la FIG. 11) para

filtrar los límites de bloque para eliminar distorsiones de efecto pixelado del vídeo reconstruido. Si se desea, el filtro de eliminación de bloques filtrará típicamente la salida del sumador 1162. También se pueden usar filtros adicionales (en el bucle o tras el bucle), además del filtro de eliminación de bloques. Dichos filtros no se muestran por razones de brevedad, pero si se desea, pueden filtrar la salida del sumador 1150 (como un filtro en el bucle).

5 [0223] Como se muestra en la FIG. 11, la memoria de datos de vídeo 1100 recibe datos de vídeo que se usan para codificar un bloque de vídeo actual dentro de una imagen de vídeo. La memoria de datos de vídeo 1100 puede almacenar datos de vídeo que se van a codificar por los componentes del codificador de vídeo 20 (por ejemplo, configurada para almacenar datos de vídeo) o almacenar datos de vídeo que se van a usar para codificar imágenes de vídeo. Los datos de vídeo almacenados en la memoria de datos de vídeo 1100 se pueden obtener, por ejemplo, a partir de la fuente de vídeo 18. La memoria de imágenes de referencia 1164 (también denominada búfer de imágenes descodificadas (DPB)) almacena datos de vídeo de referencia para su uso en la codificación de datos de vídeo por el codificador de vídeo 20 (por ejemplo, en modos de intracodificación o modos de intercodificación). La memoria de datos de vídeo 1100 y la memoria de imágenes de referencia 1164 se pueden formar por cualquiera de una variedad de dispositivos de memoria, tales como memoria dinámica de acceso aleatorio (DRAM), incluyendo DRAM síncrona (SDRAM), RAM magnetoresistiva (MRAM), RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. La memoria de datos de vídeo 1100 y el DPB 1164 se pueden proporcionar por el mismo dispositivo de memoria o por dispositivos de memoria separados. En diversos ejemplos, la memoria de datos de vídeo 1100 puede estar en un chip con otros componentes del codificador de vídeo 20, o fuera del chip en relación con esos componentes.

10 [0224] Durante el procedimiento de codificación, el codificador de vídeo 20 recibe una imagen de vídeo (por ejemplo, una trama) o un fragmento que se va a codificar. La imagen o el fragmento se pueden dividir en múltiples bloques de vídeo. La unidad de estimación de movimiento 1142 y la unidad de compensación de movimiento 1144 realizan la codificación interpredictiva del bloque de vídeo recibido en relación con uno o más bloques en una o más tramas de referencia para proporcionar predicción temporal. Además, la unidad de estimación de movimiento 1142 y la unidad de compensación de movimiento 1144 pueden realizar la codificación interpredictiva en el bloque de vídeo de recepción en relación con uno o más bloques en una o más imágenes de referencia en una vista distinta a la vista que incluye el bloque actual. La unidad de intrapredicción 1146, de forma alternativa, puede realizar la codificación intrapredictiva del bloque de vídeo recibido en relación con uno o más bloques vecinos en la misma imagen o fragmento que el bloque que se va a codificar para proporcionar predicción espacial. El codificador de vídeo 20 puede realizar múltiples pasos de codificación, (por ejemplo, para seleccionar un modo de codificación apropiado para cada bloque de datos de vídeo).

15 [0225] Además, la unidad de división 1148 puede dividir bloques de datos de vídeo en subbloques, en base a la evaluación de los esquemas de división previos en los pasos de codificación previos. Por ejemplo, la unidad de división 1148 puede dividir inicialmente una imagen o fragmento en LCU, y dividir cada una de las LCU en sub-CU, en base a un análisis de velocidad-distorsión (por ejemplo, optimización de velocidad-distorsión). La unidad de selección de modo 1140 puede producir además una estructura de datos de árbol cuaternario, indicativa de la división de una LCU en las sub-CU. Las CU de nodos hojas del árbol cuaternario pueden incluir una o más PU y una o más TU.

20 [0226] La unidad de selección de modo 1140 puede seleccionar uno de los modos de codificación, intra o inter (incluyendo entre vistas), por ejemplo, en base a resultados de error, y proporciona el bloque intra o intercodificado resultante al sumador 1150 para generar datos de bloque residuales y al sumador 1162 para reconstruir el bloque codificado para su uso como imagen de referencia. La unidad de selección de modo 1140 también proporciona elementos de sintaxis, tales como vectores de movimiento, indicadores de intramodo, información de división y otra información de sintaxis de este tipo, a la unidad de codificación por entropía 1156.

25 [0227] La unidad de estimación de movimiento 1142 y la unidad de compensación de movimiento 1144 pueden estar altamente integradas, pero se ilustran por separado para propósitos conceptuales. La estimación de movimiento, realizada por la unidad de estimación de movimiento 1142, es el procedimiento de generar vectores de movimiento (incluyendo los vectores de movimiento de disparidad), que estiman el movimiento para bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de una PU de un bloque de vídeo dentro de una trama o imagen de vídeo actual en relación con un bloque predictivo dentro de una imagen de referencia (u otra unidad codificada) en relación con el bloque actual que se está codificando dentro de la imagen actual (u otra unidad codificada). Un bloque predictivo es un bloque que se encuentra que coincide estrechamente con el bloque que se va a codificar, en términos de la diferencia de píxeles, que se puede determinar por la suma de diferencia absoluta (SAD), la suma de diferencia al cuadrado (SSD) u otras métricas de diferencia. En algunos ejemplos, el codificador de vídeo 20 puede calcular valores para posiciones de píxel fraccionario de imágenes de referencia almacenadas en la memoria de imágenes de referencia 1164. Por ejemplo, el codificador de vídeo 20 puede interpolar valores de posiciones de un cuarto de píxel, posiciones de un octavo de píxel u otras posiciones fraccionarias de píxel de la imagen de referencia. Por lo tanto, la unidad de estimación de movimiento 1142 puede realizar una búsqueda de movimiento en relación con las posiciones de píxel completo y a las posiciones de píxel fraccionario, y emitir un vector de movimiento con una precisión de píxel fraccionario.

5 [0228] La unidad de estimación de movimiento 1142 calcula un vector de movimiento para una PU de un bloque de vídeo en un fragmento intercodificado, comparando la posición de la PU con la posición de un bloque predictivo de una imagen de referencia. La imagen de referencia se puede seleccionar de una primera lista (lista 0) de imágenes de referencia o de una segunda lista (lista 1) de imágenes de referencia, de las que cada una identifica una o más imágenes de referencia almacenadas en una memoria de imágenes de referencia 1164. La unidad de estimación de movimiento 1142 envía el vector de movimiento calculado a la unidad de codificación por entropía 1156 y a la unidad de compensación de movimiento 1144.

10 [0229] La compensación de movimiento, realizada por la unidad de compensación de movimiento 1144, puede implicar buscar o generar el bloque predictivo en base al vector de movimiento determinado por la unidad de estimación de movimiento 1142. De nuevo, la unidad de estimación de movimiento 1142 y la unidad de compensación de movimiento 1144 se pueden integrar funcionalmente, en algunos ejemplos. Tras recibir el vector de movimiento para la PU del bloque de vídeo actual, la unidad de compensación de movimiento 1144 puede localizar el bloque predictivo al que apunta el vector de movimiento en una de las listas de imágenes de referencia.

15 El sumador 1150 forma un bloque de vídeo residual restando los valores de píxel del bloque predictivo a los valores de píxel del bloque de vídeo actual que se está codificando, formando valores de diferencias de píxel, como se analiza a continuación. En general, la unidad de estimación de movimiento 1142 realiza la estimación de movimiento en relación con los componentes de luma, y la unidad de compensación de movimiento 1144 usa los vectores de movimiento calculados en base a los componentes de luma tanto para los componentes de croma como para los componentes de luma. La unidad de selección de modo 1140 también puede generar elementos de sintaxis asociados con 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.

20 [0230] La unidad de intrapredicción 1146 puede intrapredicir un bloque actual, como alternativa a la interpredicción realizada por la unidad de estimación de movimiento 1142 y la unidad de compensación de movimiento 1144, como se describe anteriormente. En particular, la unidad de intrapredicción 1146 puede determinar un modo de intrapredicción a usar para codificar un bloque actual. En algunos ejemplos, la unidad de intrapredicción 1146 puede codificar un bloque actual usando diversos modos de intrapredicción, por ejemplo, durante pasos de codificación separados, y la unidad de intrapredicción 1146 (o la unidad de selección de modo 1140, en algunos ejemplos) puede seleccionar un modo de intrapredicción apropiado a usar de los modos sometidos a prueba.

25 [0231] Por ejemplo, la unidad de intrapredicción 1146 puede calcular valores de velocidad-distorsión usando un análisis de velocidad-distorsión para los diversos modos de intrapredicción sometidos a prueba, y seleccionar el modo de intrapredicción que tenga las mejores características de velocidad-distorsión entre los modos sometidos a prueba. El análisis de velocidad-distorsión determina, en general, una cantidad de distorsión (o de errores) entre un bloque codificado y un bloque original, no codificado, que se codificó para producir el bloque codificado, así como una velocidad de transferencia de bits (es decir, un número de bits) usada para producir el bloque codificado. La unidad de intrapredicción 1146 puede calcular proporciones a partir de las distorsiones y velocidades para los diversos bloques codificados, para determinar qué modo de intrapredicción presenta el mejor valor de velocidad-distorsión para el bloque.

30 [0232] Despues de seleccionar un modo de intrapredicción para un bloque, la unidad de intrapredicción 1146 puede proporcionar información, indicativa del modo de intrapredicción seleccionado para el bloque, a la unidad de codificación por entropía 1156. La unidad de codificación por entropía 1156 puede codificar la información que indica el modo de intrapredicción seleccionado. El codificador de vídeo 20 puede incluir, en el flujo de bits transmitido, datos de configuración, que pueden incluir una pluralidad de tablas de índices de modo de intrapredicción y una pluralidad de tablas de índices de modo de intrapredicción modificadas (también denominadas tablas de correlación de palabras de código), definiciones de contextos de codificación para diversos bloques e indicaciones de un modo de intrapredicción más probable, una tabla de índices de modo de intrapredicción y una tabla de índices de modo de intrapredicción modificada a usar para cada uno de los contextos.

35 [0233] El codificador de vídeo 20 forma un bloque de vídeo residual restando los datos de predicción de la unidad de selección de modo 1140 del bloque de vídeo original que se está codificando. El sumador 1150 representa el componente o los componentes que realizan esta operación de resta. La unidad de procesamiento de transformada 1152 aplica una transformada, tal como una transformada discreta de coseno (DCT) o una transformada conceptualmente similar, al bloque residual, produciendo un bloque de vídeo que comprende valores de coeficientes de transformada residuales. La unidad de procesamiento de transformada 1152 puede realizar otras transformadas que son conceptualmente similares a la DCT. También se podrían usar transformadas de ondículas, transformadas de números enteros, transformadas de subbandas u otros tipos de transformadas. En cualquier caso, la unidad de procesamiento de transformada 1152 aplica la transformada al bloque residual, produciendo un bloque de coeficientes de transformada residuales. La transformada puede convertir la información residual desde un dominio de valores de píxel a un dominio de transformada, tal como un dominio de frecuencia. La unidad de procesamiento de transformada 1152 puede enviar los coeficientes de transformada resultantes a la unidad de cuantificación 1154. La unidad de cuantificación 1154 cuantifica los coeficientes de transformada para reducir además la velocidad de transferencia de bits. El procedimiento de cuantificación puede reducir la profundidad de

bits asociada con algunos, o con la totalidad, de los coeficientes. El grado de cuantificación se puede modificar ajustando un parámetro de cuantificación. En algunos ejemplos, la unidad de cuantificación 1154 puede realizar a continuación una exploración de la matriz que incluye los coeficientes de transformada cuantificados. De forma alternativa, la unidad de codificación por entropía 1156 puede realizar la exploración.

5 [0234] Despues de la cuantificación, la unidad de codificación por entropía 1156 codifica por entropía los coeficientes de transformada cuantificados. Por ejemplo, la unidad de codificación por entropía 1156 puede realizar la codificación de longitud variable adaptativa al contexto (CAVLC), la codificación aritmética binaria adaptativa el contexto (CABAC), la codificación aritmética binaria adaptativa el contexto basada en sintaxis (SBAC), la codificación por entropía por división en intervalos de probabilidad (PIPE) u otra técnica de codificación por entropía. En el caso de la codificación por entropía basada en el contexto, el contexto se puede basar en bloques vecinos. Despues de la codificación por entropía realizada por la unidad de codificación por entropía 56, el flujo de bits codificado se puede transmitir a otro dispositivo (por ejemplo, el descodificador de vídeo 30) o guardarse para su posterior transmisión o recuperación.

10 15 [0235] La unidad de cuantificación inversa 1158 y la unidad de procesamiento de transformada inversa 1160 aplican la cuantificación inversa y la transformación inversa, respectivamente, para reconstruir el bloque residual en el dominio de píxel, por ejemplo, para su posterior uso como bloque de referencia. La unidad 1144 de compensación de movimiento puede calcular un bloque de referencia añadiendo el bloque residual a un bloque predictivo de una de las tramas de la memoria de imágenes de referencia 1164. La unidad de compensación de movimiento 1144 también puede aplicar uno o más filtros de interpolación al bloque residual reconstruido para calcular valores de píxel fraccionarios, para su uso en la estimación de movimiento. El sumador 1162 añade el bloque residual reconstruido al bloque de predicción compensado por movimiento, producido por la unidad de compensación de movimiento 1144, para producir un bloque de vídeo reconstruido para su almacenamiento en la memoria de imágenes de referencia 1164. El bloque de vídeo reconstruido se puede usar por la unidad de estimación de movimiento 1142 y la unidad de compensación de movimiento 1144, como bloque de referencia para intercodificar un bloque en una trama de vídeo posterior.

20 30 35 [0236] El codificador de vídeo 20 de la FIG. 11 representa un ejemplo de un codificador de vídeo configurado para realizar diversos procedimientos descritos en la presente divulgación. Por ejemplo, la unidad de selección de modo 1140 se puede configurar para implementar las técnicas descritas en la presente divulgación. En algunos ejemplos, la unidad de selección de modo 1140 junto con una o más de otras unidades en el codificador de vídeo 20 o una o más unidades distintas de la unidad de selección de modo 1140 se pueden configurar para implementar las técnicas de ejemplo descritas en la presente divulgación. En algunos ejemplos, un procesador (no ilustrado) del codificador de vídeo 20 o el dispositivo de origen 12 se puede configurar para implementar las técnicas de ejemplo descritas en la presente divulgación.

40 45 [0237] La FIG. 12 es un diagrama de bloques que ilustra un ejemplo de un descodificador de vídeo que puede implementar o utilizar de otro modo técnicas para la identificación de bloques de acuerdo con una o más técnicas de ejemplo descritas en la presente divulgación. En el ejemplo de la FIG. 12, el descodificador de vídeo 30 incluye una memoria de datos de vídeo 1200, una unidad de descodificación por entropía 1280 y una unidad de procesamiento de predicción 1281 que incluye una unidad de compensación de movimiento 1282, una unidad de predicción de vectores de movimiento 1283 y una unidad de intrapredicción 1284. El descodificador de vídeo 30 también incluye la unidad de cuantificación inversa 1286, la unidad de procesamiento de transformada inversa 1288, la memoria de imagen de referencia 1292 y el sumador 1290. En algunos ejemplos, el descodificador de vídeo 30 puede realizar un pase de descodificación, en general, recíproco al pase de codificación descrito con respecto al codificador de vídeo 20 (FIG. 11). Por ejemplo, el descodificador de vídeo 30 puede realizar una descodificación de intrapredicción o una descodificación de interpredicción.

50 55 [0238] En el ejemplo de la FIG. 12, la memoria de datos de vídeo 1200 recibe vídeo codificado. La memoria de datos de vídeo 1200 puede almacenar datos de vídeo (por ejemplo, configurada para almacenar datos de vídeo), tales como un flujo de bits de vídeo codificado, que se va a descodificar por los componentes del descodificador de vídeo 30B. Los datos de vídeo almacenados en la memoria de datos de vídeo 1200 se pueden obtener desde una fuente de vídeo local, tal como una cámara, por medio de la comunicación de datos de vídeo de red por cable o inalámbrica, o al acceder a medios físicos de almacenamiento de datos. La memoria de datos de vídeo 1200 puede formar un búfer de imágenes codificadas (CPB) que almacena datos de vídeo codificados a partir de un flujo de bits de vídeo codificado.

60 65 [0239] La memoria de imágenes de referencia 1292 es un ejemplo de un búfer de imágenes descodificadas (DPB) que almacena datos de vídeo de referencia para su uso en la descodificación de datos de vídeo por el descodificador de vídeo 30 (por ejemplo, en el modo de intracodificación y el modo de intercodificación). La memoria de datos de vídeo 1200 y la memoria de imágenes de referencia 1292 se pueden formar por cualquiera de una variedad de dispositivos de memoria, tales como memoria dinámica de acceso aleatorio (DRAM), incluyendo DRAM síncrona (SDRAM), RAM magnetoresistiva (MRAM), RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. La memoria de datos de vídeo 1200 y la memoria de imágenes de referencia 1292 se pueden proporcionar por el mismo dispositivo de memoria o por dispositivos de memoria separados. En diversos

ejemplos, la memoria de datos de vídeo 1200 puede estar en un chip con otros componentes del descodificador de vídeo 30, o fuera del chip en relación con esos componentes.

5       **[0240]** El codificador de vídeo 30 de la FIG. 12 representa un ejemplo de un descodificador de vídeo configurado para realizar diversos procedimientos descritos en la presente divulgación. Por ejemplo, la unidad de procesamiento de predicción 1281 se puede configurar para implementar las técnicas descritas en la presente divulgación. En algunos ejemplos, la unidad de procesamiento de predicción 1281 junto con una o más de otras unidades en el descodificador de vídeo 30 o una o más unidades distintas de la unidad de procesamiento de predicción 1281 se pueden configurar para implementar las técnicas de ejemplo descritas en la presente divulgación. En algunos ejemplos, un procesador (no ilustrado) del descodificador de vídeo 30 o el dispositivo de destino 14 se puede configurar para implementar las técnicas de ejemplo descritas en la presente divulgación.

10       **[0241]** La unidad de compensación de movimiento 1282 puede generar datos de predicción en base a vectores de movimiento recibidos desde la unidad de descodificación por entropía 1280, mientras que la unidad de intrapredicción 1284 puede generar datos de predicción en base a indicadores de modo de intrapredicción recibidos desde la unidad de descodificación por entropía 1280. La unidad de predicción de vector de movimiento 1283 puede predecir vectores de movimiento para interpredicción (por ejemplo, descodificación por interpredicción), incluyendo vectores de movimiento de disparidad.

15       **[0242]** Durante el procedimiento de descodificación, el descodificador de vídeo 30 recibe un flujo de bits de vídeo codificado que representa bloques de vídeo de un fragmento de vídeo codificado y elementos de sintaxis asociados del codificador de vídeo 20 (o a través de un dispositivo intermedio tal como el dispositivo de almacenamiento 31). La unidad de descodificación por entropía 1280 del descodificador de vídeo 30 descodifica por entropía el flujo de bits para generar coeficientes cuantificados, vectores de movimiento o indicadores de modo de intrapredicción y otros elementos de sintaxis. La unidad de descodificación por entropía 1280 reenvía los vectores de movimiento y otros elementos de sintaxis a la unidad de compensación de movimiento 1282 de la unidad de procesamiento de predicción 1281. El descodificador de vídeo 30 puede recibir los elementos de sintaxis a nivel de fragmento de vídeo y/o a nivel de bloque de vídeo.

20       **[0243]** Cuando el fragmento de vídeo se codifica como un fragmento intracodificado (I), la unidad de intrapredicción 1284 de la unidad de procesamiento de predicción 1281 puede generar datos de predicción para un bloque de vídeo del fragmento de vídeo actual, en base a un modo de intrapredicción señalizado y a datos de bloques previamente descodificados de la trama o imagen actual. Cuando la imagen de vídeo se codifica como un fragmento intercodificado (es decir, B, P en base al vector de movimiento temporal o vector de movimiento de disparidad), la unidad de compensación de movimiento 1282 produce bloques predictivos para un bloque de vídeo del fragmento de vídeo actual en base a los vectores de movimiento y otros elementos de sintaxis recibidos desde la unidad de descodificación por entropía 1280. Los bloques predictivos se pueden producir a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El descodificador de vídeo 30 puede construir las listas de imágenes de referencia, lista 0 y lista 1, usando técnicas de construcción por defecto en base a las imágenes de referencia almacenadas en la memoria de imágenes de referencia 1292. La unidad de compensación de movimiento 1282 determina la información de predicción para un bloque de vídeo del fragmento de vídeo actual, analizando los vectores de movimiento y otros elementos de sintaxis, y usa la información de predicción para producir los bloques predictivos para el bloque de vídeo actual que se está descodificando. Por ejemplo, la unidad de compensación de movimiento 1282 usa algunos de los elementos de sintaxis recibidos para determinar un modo de predicción (por ejemplo, intra o interpredicción) usado para codificar los bloques de vídeo del fragmento de vídeo, un tipo de fragmento de interpredicción (por ejemplo, fragmento B, fragmento P en base al vector de movimiento temporal o al vector de movimiento de disparidad), información de construcción para una o más de las listas de imágenes de referencia para el fragmento, vectores de movimiento para cada bloque de vídeo intercodificado del fragmento, estado de interpredicción para cada bloque de vídeo intercodificado del fragmento y otra información para descodificar los bloques de vídeo en el fragmento de vídeo actual.

25       **[0244]** La unidad de compensación de movimiento 1282 también puede realizar interpolación en base a filtros de interpolación. La unidad de compensación de movimiento 1282 puede usar filtros de interpolación como los usados por el codificador de vídeo 20 durante la codificación de los bloques de vídeo para calcular los valores interpolados para píxeles fraccionarios de los bloques de referencia. En este caso, la unidad de compensación de movimiento 1282 puede determinar los filtros de interpolación usados por el codificador de vídeo 20 a partir de los elementos de sintaxis recibidos y usar los filtros de interpolación para producir bloques predictivos.

30       **[0245]** La unidad de cuantificación inversa 1286 cuantifica inversamente, es decir, descuantifica, los coeficientes de transformada cuantificados proporcionados en el flujo de bits y descodificados por la unidad de descodificación por entropía 1280. El procedimiento de cuantificación inversa puede incluir el uso de un parámetro de cuantificación QPy, calculado por el descodificador de vídeo 30 para cada bloque de vídeo en el fragmento de vídeo, para determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa que se debería aplicar. La unidad de procesamiento de transformada inversa 1288 aplica una transformada inversa, por ejemplo, una DCT inversa, una transformada de entero inversa o un procedimiento de transformada inversa conceptualmente similar, a los coeficientes de transformada para producir bloques residuales en el dominio de píxel.

5 [0246] Después de que la unidad de compensación de movimiento 1282 genera el bloque predictivo para el bloque de vídeo actual en base a los vectores de movimiento y otros elementos de sintaxis, el descodificador de vídeo 30 forma un bloque de vídeo descodificado sumando los bloques residuales de la unidad de transformada inversa 1288 a los bloques predictivos correspondientes generados por la unidad de compensación de movimiento 1282. El sumador 1290 representa el componente o los componentes que realizan esta operación de suma. Si se desea, también se puede aplicar un filtro de eliminación de bloques para filtrar los bloques descodificados para eliminar distorsiones de efecto pixelado. También se pueden usar otros filtros de bucle (en el bucle de codificación o bien después del bucle de codificación) para suavizar las transiciones de píxeles o mejorar de otro modo la calidad del vídeo. Los bloques de vídeo descodificados en una trama o imagen dada se almacenan a continuación en la memoria de imágenes de referencia 1292, que almacena imágenes de referencia usadas para la compensación de movimiento posterior. La memoria de imágenes de referencia 1292 también almacena vídeo descodificado para su presentación posterior en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la FIG. 1.

10 15 [0247] En la presente divulgación, el codificador de vídeo 20 y el descodificador de vídeo 30 se pueden denominar genéricamente codificador de vídeo. Por ejemplo, un dispositivo para codificación de vídeo (por ejemplo, el dispositivo de origen 12, el dispositivo de destino 14, un microprocesador o un circuito integrado (IC) incluye una memoria de datos de vídeo (por ejemplo, memoria de datos de vídeo 1100 o memoria de datos de vídeo 1200) configurada para almacenar datos de vídeo. El codificador de vídeo (por ejemplo, el codificador de vídeo 20 o el descodificador de vídeo 30) incluye uno o más procesadores y está configurado para determinar un vector de disparidad para un bloque actual en una imagen actual en una vista actual en base a los datos de vídeo almacenados. Por ejemplo, los datos de vídeo almacenados pueden incluir información de movimiento del bloque vecino, y el codificador de vídeo puede determinar si alguno de los bloques vecinos se interpreda con un vector de movimiento de disparidad o si existe un IDV para un bloque vecino en base a los datos de vídeo almacenado. El codificador de vídeo puede determinar el vector de disparidad para el bloque actual en base a un vector de movimiento de disparidad o un IDV de un bloque vecino.

20 30 35 [0248] El codificador de vídeo puede determinar un bloque en una imagen de referencia en una vista de referencia en base a una localización en la imagen de referencia a la que se refiere el vector de disparidad en base al vector de disparidad comenzando desde un píxel inferior derecho en un subbloque 2x2 central dentro del bloque actual. El codificador de vídeo puede codificar (por ejemplo, codificar o descodificar) por interprediccción el bloque actual en base al bloque determinado. Por ejemplo, el codificador de vídeo puede utilizar la información de movimiento del bloque determinado si el bloque determinado se va a usar para la codificación por interprediccción del bloque actual. Como otro ejemplo, el codificador de vídeo puede utilizar valores de píxel de una reconstrucción del bloque determinado para determinar un bloque residual para el bloque actual. En estos ejemplos, el vector de disparidad se puede considerar como un vector de movimiento de disparidad.

40 45 50 [0249] La FIG. 13 es un diagrama de flujo que ilustra un procedimiento de ejemplo de descodificación de vídeo. Como se ilustra, el descodificador de vídeo 30 puede determinar un vector de disparidad para un bloque actual en una imagen actual en una vista actual (1300). Puede haber diversos modos en los que el descodificador de vídeo 30 puede determinar el vector de disparidad, incluyendo las técnicas de derivación de NBDV y el refinamiento adicional del vector de disparidad derivado. Por ejemplo, el descodificador de vídeo 30 puede comprobar la información de movimiento de los bloques vecinos (bloque vecino espacial o temporal) como se almacena en la memoria de datos de vídeo 1200 para determinar si alguno de los bloques vecinos se interpreda con un vector de movimiento de disparidad o existe un IDV para un bloque vecino. El descodificador de vídeo 30 puede convertir el vector de movimiento de disparidad de un bloque vecino o un IDV de bloque vecino en el vector de disparidad del bloque actual, y puede refinarse adicionalmente el vector de disparidad para determinar un nuevo vector de disparidad. En este ejemplo, el vector de disparidad se refiere al vector de disparidad sin refinarse o al vector de disparidad refinado.

55 60 [0250] El descodificador de vídeo 30 puede determinar un bloque en una imagen de referencia en una vista de referencia en base a una localización en la imagen de referencia a la que se refiere el vector de disparidad en base al vector de disparidad comenzando desde un píxel inferior derecho en un subbloque 2x2 central dentro del bloque actual (1302). Por ejemplo, la FIG. 4 ilustra el bloque actual 38 que incluye un subbloque 48 dentro del centro del bloque actual 38. Debido a que al menos uno de la altura y el ancho del bloque actual 38 es par, no hay un único píxel central en el bloque actual 38. Más bien, cuatro píxeles que forman el subbloque 48 están dentro del centro del bloque actual 38. Para determinar el bloque en la imagen de referencia 34, el descodificador de vídeo 30 puede comenzar el vector de disparidad desde el píxel inferior derecho en el subbloque 44 (por ejemplo, la posición 46).

65 [0251] Como se ilustra, el vector de disparidad DV2 comienza desde la posición 46 y se refiere a la localización C3 en la imagen de referencia 34. El descodificador de vídeo 30 puede determinar un bloque 4x4 que cubre la localización en la imagen de referencia a la que se refiere el vector de disparidad. Por ejemplo, el descodificador de vídeo 30 puede determinar que el bloque 42 cubre la localización C3 en la imagen de referencia 34 a la que se refiere el vector de disparidad DV2.

**[0252]** En algunos ejemplos, para determinar el bloque en la imagen de referencia, el descodificador de vídeo 30 puede determinar una coordenada x y una coordenada y para la localización en la imagen de referencia. La coordenada x para la localización en la imagen de referencia es igual a Clip3(0, PicWidthInSamples<sub>L</sub> - 1, xP + ((nPSW) >> 1) + ((mvDisp[0] + 2) >> 2)), y la coordenada y para la localización en la imagen de referencia es igual a Clip3(0, PicHeightInSamples<sub>L</sub> - 1, yP + ((nPSH) >> 1) + ((mvDisp[1] + 2) >> 2)). PicWidthInSamples<sub>L</sub> es igual a un ancho de la imagen de referencia y PicHeightInSamples<sub>L</sub> es igual a una altura de la imagen de referencia. xP es igual a una coordenada x de una esquina superior izquierda del bloque actual y yP es igual a una coordenada y de una esquina superior izquierda del bloque actual. nPSW es igual a un ancho del bloque actual y nPSH es igual a una altura del bloque actual. mvDisp[0] es igual a un componente x del vector de disparidad y mvDisp[1] es igual a un componente y del vector de disparidad. Por ejemplo, una coordenada x para el píxel inferior derecho en el subbloque 2x2 central dentro del bloque actual es igual a xP + ((nPSW) >> 1), y una coordenada y para el píxel inferior derecho en el subbloque 2x2 central dentro del bloque actual es igual a yP + ((nPSH) >> 1).

**[0253]** El descodificador de vídeo 30 puede descodificar por interprediccción el bloque actual en base al bloque determinado (1304). Por ejemplo, el descodificador de vídeo 30 puede descodificar por interprediccción el bloque actual usando una de predicción de movimiento entre vistas o predicción residual entre vistas si se determina que el bloque determinado se va a usar para descodificación por interprediccción. Como un ejemplo, el descodificador de vídeo 30 puede formar una lista de predictores de vectores de movimiento candidatos y puede insertar información de movimiento del bloque determinado en la lista de predictores de vectores de movimiento candidatos. En este ejemplo, el descodificador de vídeo 30 puede descodificar por interprediccción el bloque actual usando la información de movimiento del bloque determinado si la información de movimiento del bloque determinado se selecciona de la lista de predictores de vectores de movimiento candidatos (por ejemplo, un índice en la lista de predictores de vectores de movimiento candidatos se refiere a la información de movimiento del bloque al que se hace referencia por el vector de disparidad en base al vector de disparidad comenzando desde un píxel inferior derecho del subbloque 2x2 central dentro del bloque actual).

**[0254]** Como otro ejemplo, el descodificador de vídeo 30 puede utilizar una reconstrucción del bloque determinado como un bloque predictivo. En este ejemplo, el descodificador de vídeo 30 recibe valores residuales que el descodificador de vídeo 30 añade al bloque determinado para descodificar por interprediccción el bloque actual. Debido a que el vector de disparidad se refiere a un bloque predictivo (por ejemplo, el bloque de referencia funciona como un bloque predictivo), el vector de disparidad se puede considerar como un vector de movimiento de disparidad.

**[0255]** La FIG. 14 es un diagrama de flujo que ilustra un procedimiento de ejemplo de codificación de vídeo. Como se ilustra, el codificador de vídeo 20 puede determinar un vector de disparidad para un bloque actual en una imagen actual en una vista actual (1400). Puede haber diversos modos en los que el codificador de vídeo 20 puede determinar el vector de disparidad, incluyendo las técnicas de derivación de NBDV y el refinamiento adicional del vector de disparidad derivado. Por ejemplo, el codificador de vídeo 20 puede comprobar la información de movimiento de los bloques vecinos (bloque vecino espacial o temporal) como se almacena en la memoria de datos de vídeo 1300 para determinar si alguno de los bloques vecinos se interpredice con un vector de movimiento de disparidad o existe un IDV para un bloque vecino. El codificador de vídeo 20 puede convertir el vector de movimiento de disparidad de un bloque vecino o un IDV de bloque vecino en el vector de disparidad del bloque actual, y puede refinar adicionalmente el vector de disparidad para determinar un nuevo vector de disparidad. En este ejemplo, el vector de disparidad se refiere a un vector de disparidad sin refinar o al vector de disparidad refinado. En general, en la presente divulgación, incluyendo el ejemplo anterior ilustrado en la FIG. 13, se usa un vector de disparidad para referirse a un vector de disparidad sin refinar o un vector de disparidad refinado.

**[0256]** El codificador de vídeo 20 puede determinar un bloque en una imagen de referencia en una vista de referencia en base a una localización en la imagen de referencia a la que se refiere el vector de disparidad en base al vector de disparidad comenzando desde un píxel inferior derecho en un subbloque 2x2 central dentro del bloque actual (1402). Por ejemplo, la FIG. 4 ilustra el bloque actual 38 que incluye un subbloque 48 dentro del centro del bloque actual 38. Debido a que al menos uno de la altura y el ancho del bloque actual 38 es par, no hay un único píxel central en el bloque actual 38. Más bien, cuatro píxeles que forman el subbloque 48 están dentro del centro del bloque actual 38. Para determinar el bloque en la imagen de referencia 34, el codificador de vídeo 20 puede comenzar el vector de disparidad desde el píxel inferior derecho en el subbloque 44 (por ejemplo, la posición 46).

**[0257]** Como se ilustra, el vector de disparidad DV2 comienza desde la posición 46 y se refiere a la localización C3 en la imagen de referencia 34. El codificador de vídeo 20 puede determinar un bloque 4x4 que cubre la localización en la imagen de referencia a la que se refiere el vector de disparidad. Por ejemplo, el codificador de vídeo 20 puede determinar que el bloque 42 cubre la localización C3 en la imagen de referencia 34 a la que se refiere el vector de disparidad DV2.

**[0258]** En algunos ejemplos, para determinar el bloque en la imagen de referencia, el codificador de vídeo 20 puede determinar una coordenada x y una coordenada y para la localización en la imagen de referencia. La coordenada x para la localización en la imagen de referencia es igual a Clip3(0, PicWidthInSamples<sub>L</sub> - 1, xP + ((nPSW) >> 1) + ((mvDisp[0] + 2) >> 2)), y la coordenada y para la localización en la imagen de referencia es igual

5 a Clip3(0, PicHeightInSamples<sub>L</sub> - 1, yP + ((nPSH) >> 1) + ((mvDisp[1] + 2) >> 2)). PicWidthInSamples<sub>L</sub> es igual a un ancho de la imagen de referencia y PicHeightInSamples<sub>L</sub> es igual a una altura de la imagen de referencia. xP es igual a una coordenada x de una esquina superior izquierda del bloque actual y yP es igual a una coordenada y de una esquina superior izquierda del bloque actual. nPSW es igual a un ancho del bloque actual y nPSH es igual a una altura del bloque actual. mvDisp[0] es igual a un componente x del vector de disparidad y mvDisp[1] es igual a un componente y del vector de disparidad. Por ejemplo, una coordenada x para el píxel inferior derecho en el subbloque 2x2 central dentro del bloque actual es igual a xP + ((nPSW) >> 1), y una coordenada y para el píxel inferior derecho en el subbloque 2x2 central dentro del bloque actual es igual a yP + ((nPSH) >> 1).

10 10 [0259] El codificador de vídeo 20 puede codificar por interpredicción el bloque actual en base al bloque determinado (1404). Por ejemplo, el codificador de vídeo 20 puede codificar por interpredicción el bloque actual usando una predicción de movimiento entre vistas o una predicción residual entre vistas si se determina que el bloque determinado se va a usar para la codificación por interpredicción. Como un ejemplo, el codificador de vídeo 20 puede formar una lista de predictores de vectores de movimiento candidatos y puede insertar información de movimiento del bloque determinado en la lista de predictores de vectores de movimiento candidatos. En este ejemplo, el codificador de vídeo 20 puede codificar por interpredicción el bloque actual usando la información de movimiento del bloque determinado si la información de movimiento del bloque determinado se selecciona de la lista de predictores de vectores de movimiento candidatos (por ejemplo, un índice en la lista de predictores de vectores de movimiento candidatos se refiere a la información de movimiento del bloque al que se hace referencia por el vector de disparidad en base al vector de disparidad comenzando desde un píxel inferior derecho del subbloque 2x2 central dentro del bloque actual).

15 20 [0260] Como otro ejemplo, el codificador 20 de vídeo puede utilizar una reconstrucción del bloque determinado como bloque predictivo. En este ejemplo, el codificador de vídeo 20 determina valores residuales entre el bloque determinado y el bloque actual y emite valores residuales. Debido a que el vector de disparidad se refiere a un bloque predictivo (por ejemplo, el bloque de referencia funciona como un bloque predictivo), el vector de disparidad se puede considerar como un vector de movimiento de disparidad.

25 30 [0261] Se ha de reconocer que, dependiendo del ejemplo, determinadas acciones o acontecimientos de cualquiera de las técnicas descritas en el presente documento se pueden realizar en una secuencia distinta, se pueden añadir, fusionar o excluir por completo (por ejemplo, no todas las acciones o acontecimientos descritos son necesarios para la puesta en práctica de las técnicas). Además, en determinados ejemplos, las acciones o acontecimientos se pueden realizar simultáneamente, por ejemplo, a través de procesamiento de múltiples subprocessos, procesamiento de interrupciones o múltiples procesadores, en lugar de secuencialmente.

35 40 [0262] En uno o más ejemplos, las funciones descritas se pueden implementar en hardware, software, firmware o en cualquier combinación de los mismos. Si se implementan en software, las funciones se pueden almacenar en, o transmitir sobre, un medio legible por ordenador como una o más instrucciones o código, y ejecutar por una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible tal como medios de almacenamiento de datos, o medios de comunicación incluyendo cualquier medio que facilita la transferencia de un programa informático de un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De este modo, los medios legibles por ordenador pueden corresponder, en general, a (1) medios de almacenamiento tangibles legibles por ordenador que sean no transitorios o a (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que se pueda acceder por uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en la presente divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

50 55 [0263] A modo de ejemplo, y no de limitación, dichos medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento en disco óptico, almacenamiento en disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que se pueda usar para almacenar el código de programa deseado en forma de instrucciones o estructuras de datos y al que se pueda acceder por un ordenador. Además, cualquier conexión recibe apropiadamente la denominación de medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, servidor u otra fuente remota usando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas están incluidos en la definición de medio. Sin embargo, se debe entender que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que, en cambio, están dirigidos a medios de almacenamiento tangibles no transitorios. Los discos, como se usan en el presente documento, incluyen disco compacto (CD), disco láser, disco óptico, disco versátil digital (DVD), disco flexible y disco Blu-ray, donde unos discos reproducen normalmente los datos de forma magnética, mientras que otros discos reproducen los datos de forma óptica con láseres. Las combinaciones de los anteriores también se deben incluir dentro del alcance de los medios legibles por ordenador.

5 [0264] Las instrucciones se pueden ejecutar por uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), matrices lógicas programables *in situ* (FPGA) u otros circuitos lógicos integrados o discretos equivalentes. Por consiguiente, el término "procesador", como se usa en el presente documento, se puede referir a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento se puede proporcionar en módulos de hardware y/o software dedicados configurados para la codificación y la descodificación, o incorporarse en un códec combinado. Además, las técnicas se podrían implementar por completo en uno o más circuitos o elementos lógicos.

10 [0265] Las técnicas de la presente divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluyendo un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En la presente divulgación se describen diversos componentes, módulos o unidades para destacar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no se requiere necesariamente su realización por diferentes unidades de hardware. En cambio, como se describe anteriormente, diversas unidades se pueden combinar en una unidad de hardware de códec o proporcionar por un grupo de unidades de hardware interoperativas, que incluye uno o más procesadores como se describe anteriormente, junto con software y/o firmware adecuados.

## REIVINDICACIONES

1. Un procedimiento de descodificación de vídeo, comprendiendo el procedimiento:
  - 5 determinar un vector de disparidad para un bloque actual en una imagen actual en una vista actual;
  - 10 determinar un primer bloque de referencia en una imagen de referencia en una vista de referencia en base a una localización en la imagen de referencia a la que se refiere el vector de disparidad en base al vector de disparidad comenzando desde un píxel inferior derecho en un subbloque 2x2 central dentro del bloque actual;
  - 15 si el primer bloque de referencia incluye información de movimiento, descodificar por interpredicción el bloque actual usando la información de movimiento para el primer bloque de referencia; y
  - 20 si el primer bloque de referencia no incluye información de movimiento, determinar un segundo bloque de referencia en la imagen de referencia en la vista de referencia en base a una localización en la imagen de referencia a la que se refiere el vector de disparidad en base al vector de disparidad comenzando desde un píxel superior izquierdo en el subbloque 2x2 central dentro del bloque actual, y si el segundo bloque de referencia incluye información de movimiento, descodificar por interpredicción el bloque actual usando la información de movimiento para el segundo bloque de referencia.
2. Un procedimiento de codificación de vídeo, comprendiendo el procedimiento:
  - 25 determinar un vector de disparidad para un bloque actual en una imagen actual en una vista actual;
  - 30 determinar un primer bloque de referencia en una imagen de referencia en una vista de referencia en base a una localización en la imagen de referencia a la que se refiere el vector de disparidad en base al vector de disparidad comenzando desde un píxel inferior derecho en un subbloque 2x2 central dentro del bloque actual;
  - 35 si el primer bloque de referencia incluye información de movimiento, codificar por interpredicción el bloque actual usando la información de movimiento para el primer bloque de referencia; y
  - 40 si el primer bloque de referencia no incluye información de movimiento, determinar un segundo bloque de referencia en la imagen de referencia en la vista de referencia en base a una localización en la imagen de referencia a la que se refiere el vector de disparidad en base al vector de disparidad comenzando desde un píxel superior izquierdo en el subbloque 2x2 central dentro del bloque actual, y si el segundo bloque de referencia incluye información de movimiento, codificar por interpredicción el bloque actual usando la información de movimiento para el segundo bloque de referencia.
3. El procedimiento de la reivindicación 1 o la reivindicación 2, en el que las coordenadas del píxel inferior derecho en el subbloque 2x2 central dentro del bloque actual son iguales a  $((xP + (nPSW >> 1)), (yP + (nPSH >> 1)))$ , en las que  $yP$  es igual a una coordenada y de una esquina superior izquierda del bloque actual, en las que  $xP$  es igual a una coordenada x de una esquina superior izquierda del bloque actual, en las que  $nPSW$  es igual a un ancho del bloque actual y en las que  $nPSH$  es igual a una altura del bloque actual.
4. El procedimiento de la reivindicación 1 o la reivindicación 2, en el que determinar el bloque en la imagen de referencia en la vista de referencia comprende:
  - 50 determinar una coordenada x y una coordenada y para la localización en la imagen de referencia; y
  - 55 determinar el bloque que cubre las coordenadas x e y determinadas, en el que la coordenada x para la localización en la imagen de referencia es igual a  $xP + ((nPSW >> 1) + ((mvDisp[0] + 2) >> 2))$ , en el que la coordenada y para la localización en la imagen de referencia es igual a  $yP + ((nPSH >> 1) + ((mvDisp[1] + 2) >> 2))$ , en las que  $xP$  es igual a una coordenada x de una esquina superior izquierda del bloque actual, en las que  $yP$  es igual a una coordenada y de una esquina superior izquierda del bloque actual, en las que  $nPSW$  es igual a un ancho del bloque actual, en las que  $nPSH$  es igual a una altura del bloque actual, en las que  $mvDisp[0]$  es igual a un componente x del vector de disparidad y en las que  $mvDisp[1]$  es igual a un componente y del vector de disparidad.
  - 60 5. El procedimiento de la reivindicación 1 o la reivindicación 2, en el que determinar el bloque en la imagen de referencia en la vista de referencia comprende:
    - 65 determinar una coordenada x y una coordenada y para la localización en la imagen de referencia; y

- determinar el bloque que cubre las coordenadas x e y determinadas,
- 5 en el que la coordenada x para la localización en la imagen de referencia es igual a Clip3(0, PicWidthInSamples<sub>L</sub> - 1, xP + ((nP SW) >> 1) + ((mvDisp[0] + 2) >> 2)), en el que la coordenada y para la localización en la imagen de referencia es igual a Clip3(0, PicHeightInSamples<sub>L</sub> - 1, yP + ((nP SH) >> 1) + ((mvDisp[1] + 2) >> 2)), en las que PicWidthInSamples<sub>L</sub> es igual a un ancho de la imagen de referencia, en las que PicHeightInSamples<sub>L</sub> es igual a una altura de la imagen de referencia, en las que xP es igual a una coordenada x de una esquina superior izquierda del bloque actual, en las que yP es igual a una coordenada y de una esquina superior izquierda del bloque actual, en las que nP SW es igual a un ancho del bloque actual, en las que nP SH es igual a una altura del bloque actual, en las que mvDisp[0] es igual a un componente x del vector de disparidad y en las que mvDisp[1] es igual a un componente y del vector de disparidad.
- 10
- 15 6. El procedimiento de la reivindicación 1 o la reivindicación 2, en el que determinar el bloque en la imagen de referencia comprende determinar un bloque 4x4 que cubre la localización en la imagen de referencia a la que se refiere el vector de disparidad.
- 20 7. El procedimiento de la reivindicación 1 o la reivindicación 2, que comprende además:
- 25 insertar información de movimiento del bloque determinado en una lista de predictores de vectores de movimiento candidatos, en el que codificar por interpredicción el bloque actual comprende codificar por interpredicción el bloque actual usando la información de movimiento del bloque determinado si la información de movimiento del bloque determinado se selecciona de la lista de predictores de vectores de movimiento candidatos.
- 30 8. El procedimiento de una cualquiera de las reivindicaciones precedentes, en el que si el segundo bloque de referencia no incluye información de movimiento, determinar un tercer bloque de referencia en la imagen de referencia en la vista de referencia en base a una localización en la imagen de referencia a la que se refiere el vector de disparidad en base al vector de disparidad comenzando desde un pixel superior derecho o inferior izquierdo en el subbloque 2x2 central dentro del bloque actual, y si el tercer bloque de referencia incluye información de movimiento, descodificar por interpredicción el bloque actual usando la información de movimiento para el tercero bloque de referencia
- 35 9. Un dispositivo para descodificación de vídeo, comprendiendo el dispositivo:
- 40 una memoria de datos de vídeo configurada para almacenar datos de vídeo; y un descodificador de vídeo que comprende uno o más procesadores y configurado para:
- 45 determinar un vector de disparidad para un bloque actual en una imagen actual en una vista actual en base a los datos de vídeo almacenados;
- 50 determinar un primer bloque de referencia en una imagen de referencia en una vista de referencia en base a una localización en la imagen de referencia a la que se refiere el vector de disparidad en base al vector de disparidad comenzando desde un pixel inferior derecho en un subbloque 2x2 central dentro del bloque actual;
- 55 si el primer bloque de referencia incluye información de movimiento, descodificar por interpredicción el bloque actual usando la información de movimiento para el primer bloque de referencia; y si el primer bloque de referencia no incluye información de movimiento, determinar un segundo bloque de referencia en la imagen de referencia en la vista de referencia en base a una localización en la imagen de referencia a la que se refiere el vector de disparidad en base al vector de disparidad comenzando desde un pixel superior izquierdo en el subbloque 2x2 central dentro del bloque actual, y si el segundo bloque de referencia incluye información de movimiento, descodificar por interpredicción el bloque actual usando la información de movimiento para el segundo bloque de referencia.
- 60 10. El dispositivo de la reivindicación 9, en el que las coordenadas del pixel inferior derecho en el subbloque 2x2 central dentro del bloque actual son iguales a ((xP + (nP SW >> 1)), (yP + (nP SH >> 1))), en las que yP es igual a una coordenada y de una esquina superior izquierda del bloque actual, en las que xP es igual a una coordenada x de una esquina superior izquierda del bloque actual, en las que nP SW es igual a un ancho del bloque actual y en las que nP SH es igual a una altura del bloque actual.
- 65

11. El dispositivo de la reivindicación 9, en el que para determinar el bloque en la imagen de referencia en la vista de referencia, el descodificador de vídeo está configurado para:

5 determinar una coordenada x y una coordenada y para la localización en la imagen de referencia; y

determinar el bloque que cubre las coordenadas x e y determinadas,

10 en el que la coordenada x para la localización en la imagen de referencia es igual a  $xP + ((nPSW) >> 1) + ((mvDisp[0] + 2) >> 2)$ , en el que la coordenada y para la localización en la imagen de referencia es igual a  $yP + ((nPSH) >> 1) + ((mvDisp[1] + 2) >> 2)$ , en las que  $xP$  es igual a una coordenada x de una esquina superior izquierda del bloque actual, en las que  $yP$  es igual a una coordenada y de una esquina superior izquierda del bloque actual, en las que  $nPSW$  es igual a un ancho del bloque actual, en las que  $nPSH$  es igual a una altura del bloque actual, en las que  $mvDisp[0]$  es igual a un componente x del vector de disparidad y en las que  $mvDisp[1]$  es igual a un componente y del vector de disparidad.

- 15 12. El dispositivo de la reivindicación 9, en el que para determinar el bloque en la imagen de referencia en la vista de referencia, el descodificador de vídeo está configurado para:

20 determinar una coordenada x y una coordenada y para la localización en la imagen de referencia; y

determinar el bloque que cubre las coordenadas x e y determinadas,

25 en el que la coordenada x para la localización en la imagen de referencia es igual a  $Clip3(0, PicWidthInSamples_L - 1, xP + ((nPSW) >> 1) + ((mvDisp[0] + 2) >> 2))$ , en el que la coordenada y para la localización en la imagen de referencia es igual a  $Clip3(0, PicHeightInSamples_L - 1, yP + ((nPSH) >> 1) + ((mvDisp[1] + 2) >> 2))$ , en las que  $PicWidthInSamples_L$  es igual un ancho de la imagen de referencia, en las que  $PicHeightInSamples_L$  es igual a una altura de la imagen de referencia, en las que  $xP$  es igual a una coordenada x de una esquina superior izquierda del bloque actual, en las que  $yP$  es igual a una coordenada y de una esquina superior izquierda del bloque actual, en las que  $nPSW$  es igual a un ancho del bloque actual, en las que  $nPSH$  es igual a una altura del bloque actual, en las que  $mvDisp[0]$  es igual a un componente x del vector de disparidad y en las que  $mvDisp[1]$  es igual a un componente y del vector de disparidad.

- 30 35 13. El dispositivo de la reivindicación 9, en el que para determinar el bloque en la imagen de referencia, el descodificador de vídeo está configurado para determinar un bloque 4x4 que cubre la localización en la imagen de referencia a la que se refiere el vector de disparidad.

- 35 40 14. El dispositivo de la reivindicación 9, en el que el dispositivo comprende uno de:

un microprocesador;

un circuito integrado; o

un dispositivo de comunicación inalámbrica que incluye el descodificador de vídeo.

- 45 45 15. Un medio de almacenamiento legible por ordenador que tiene instrucciones almacenadas en el mismo que cuando se ejecutan hacen que uno o más procesadores de un dispositivo para codificación de vídeo realicen el procedimiento de acuerdo con una cualquiera de las reivindicaciones 1 a 8.

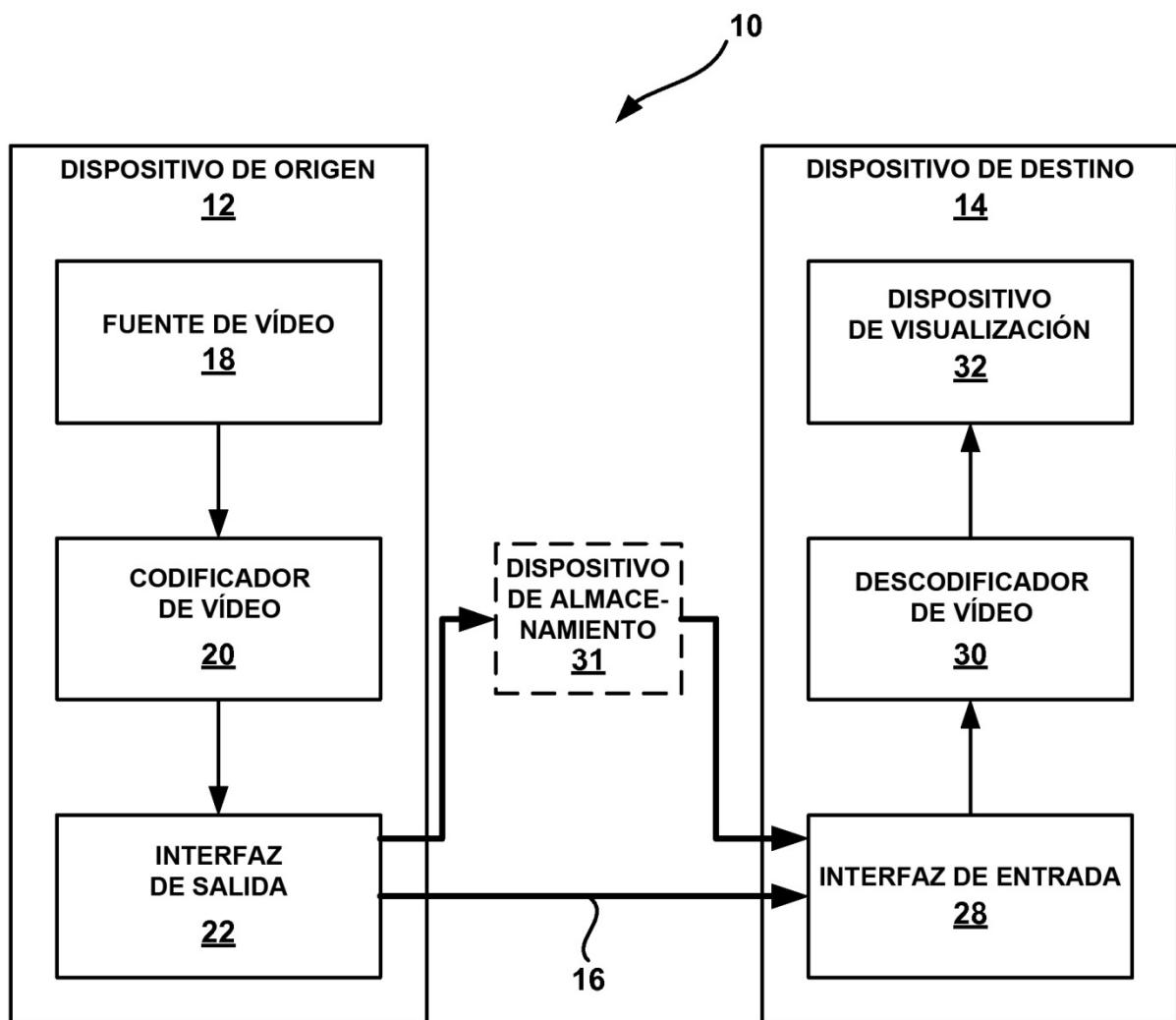
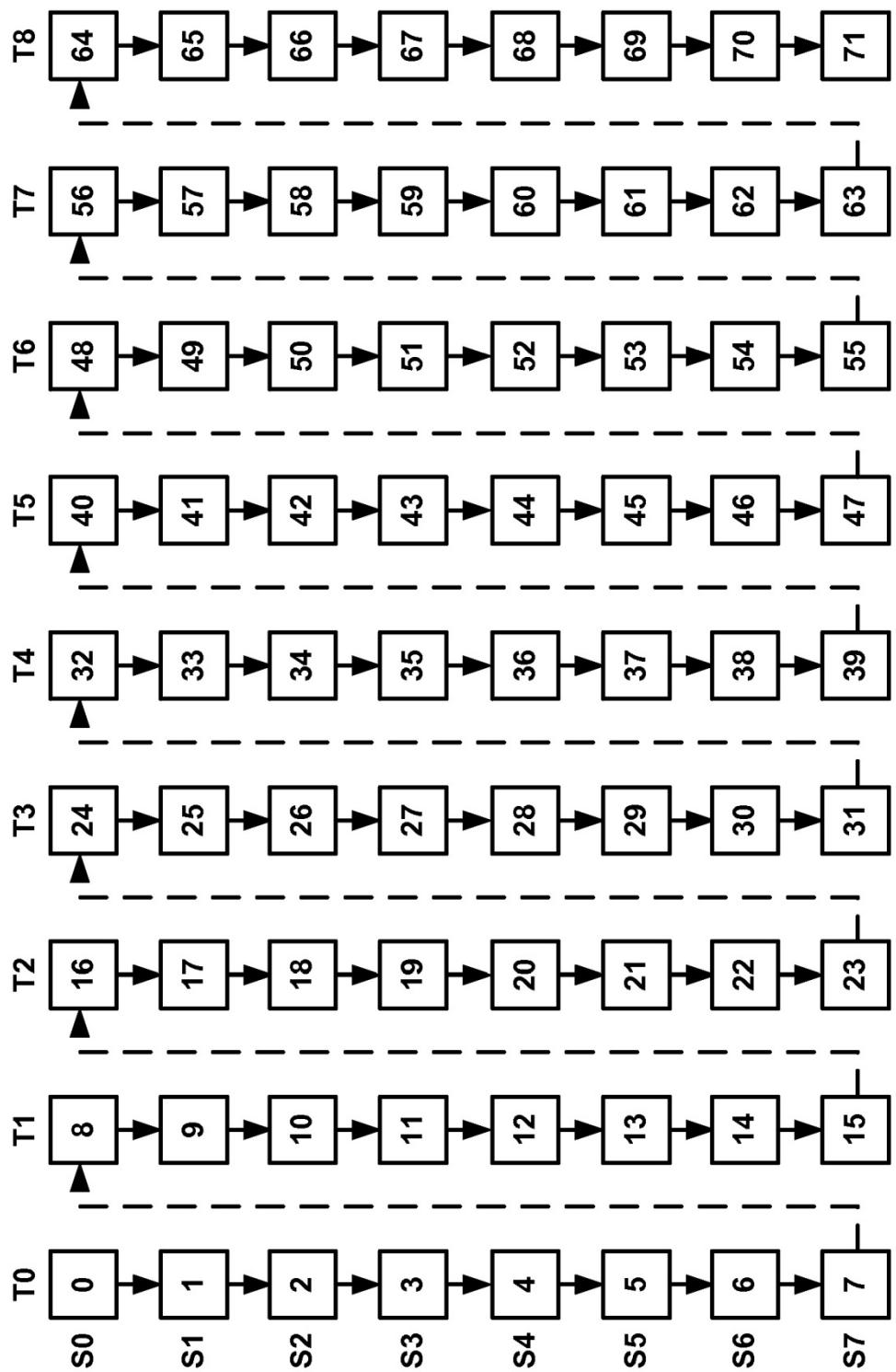
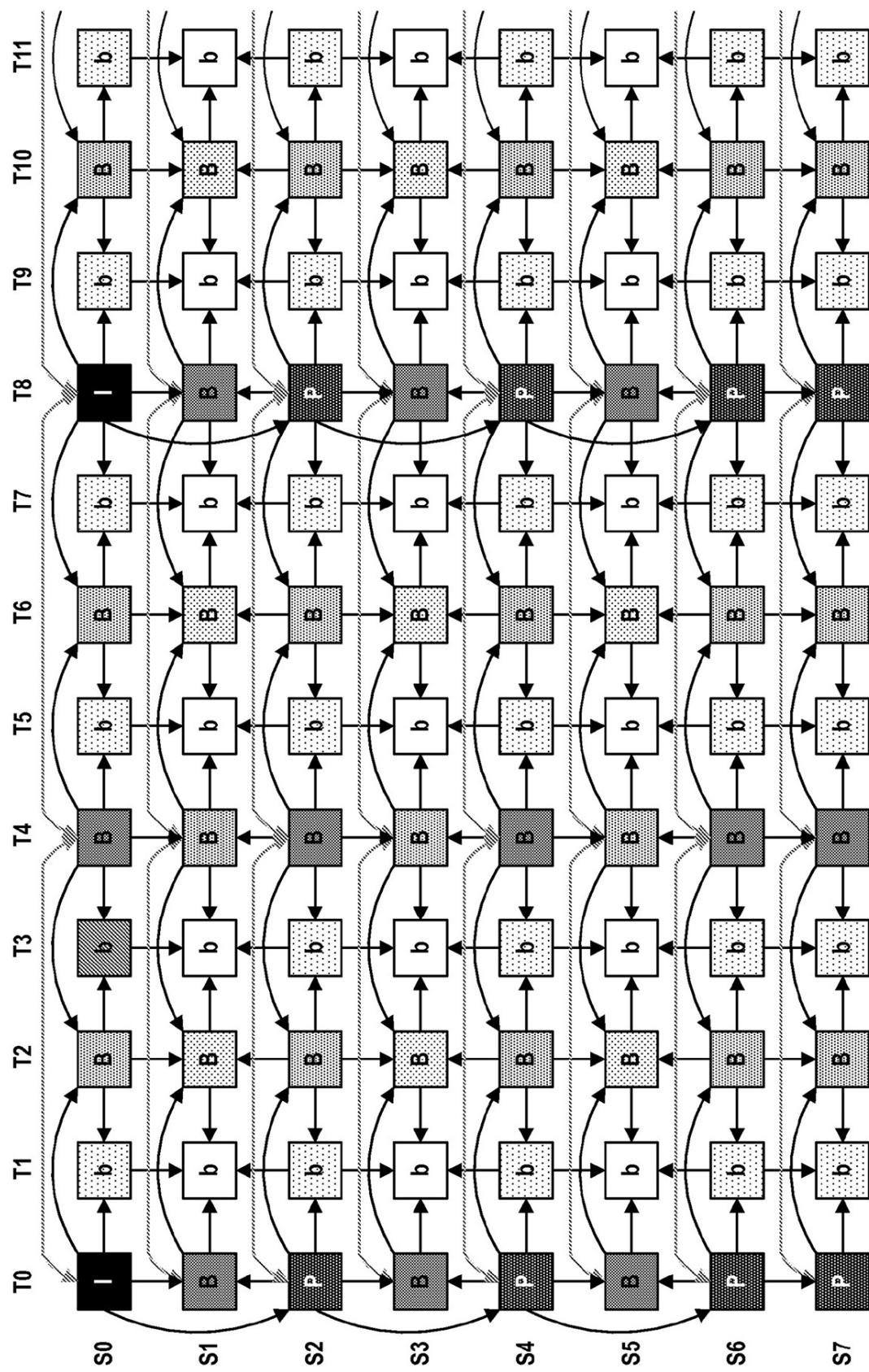


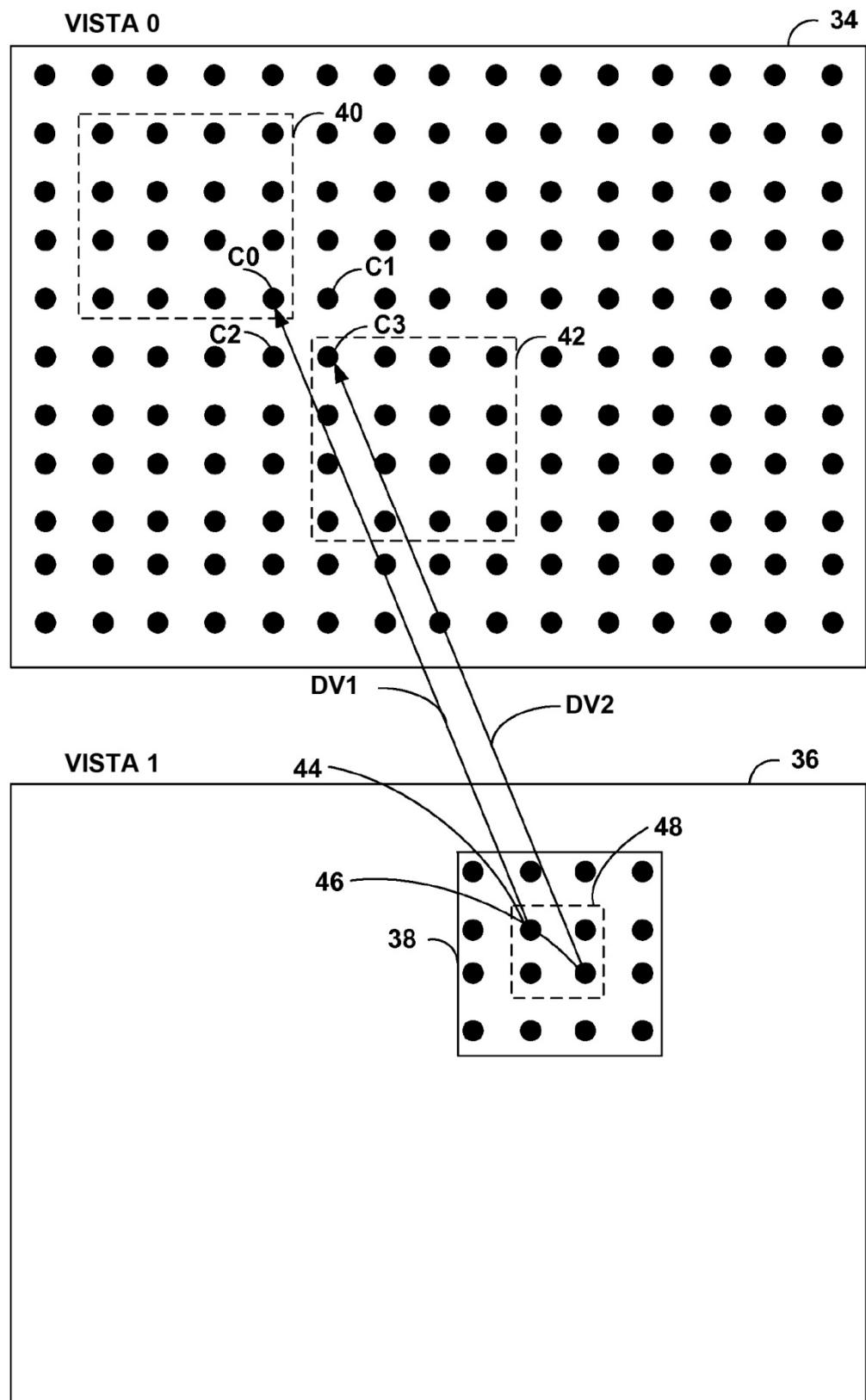
FIG. 1

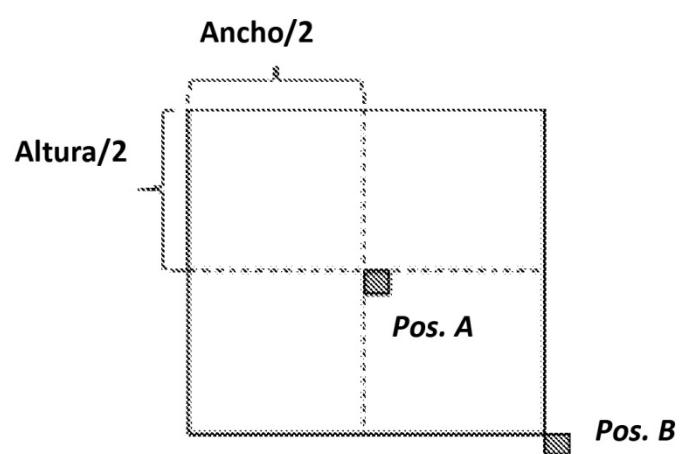


**FIG. 2**  
Codificación de tiempo primero

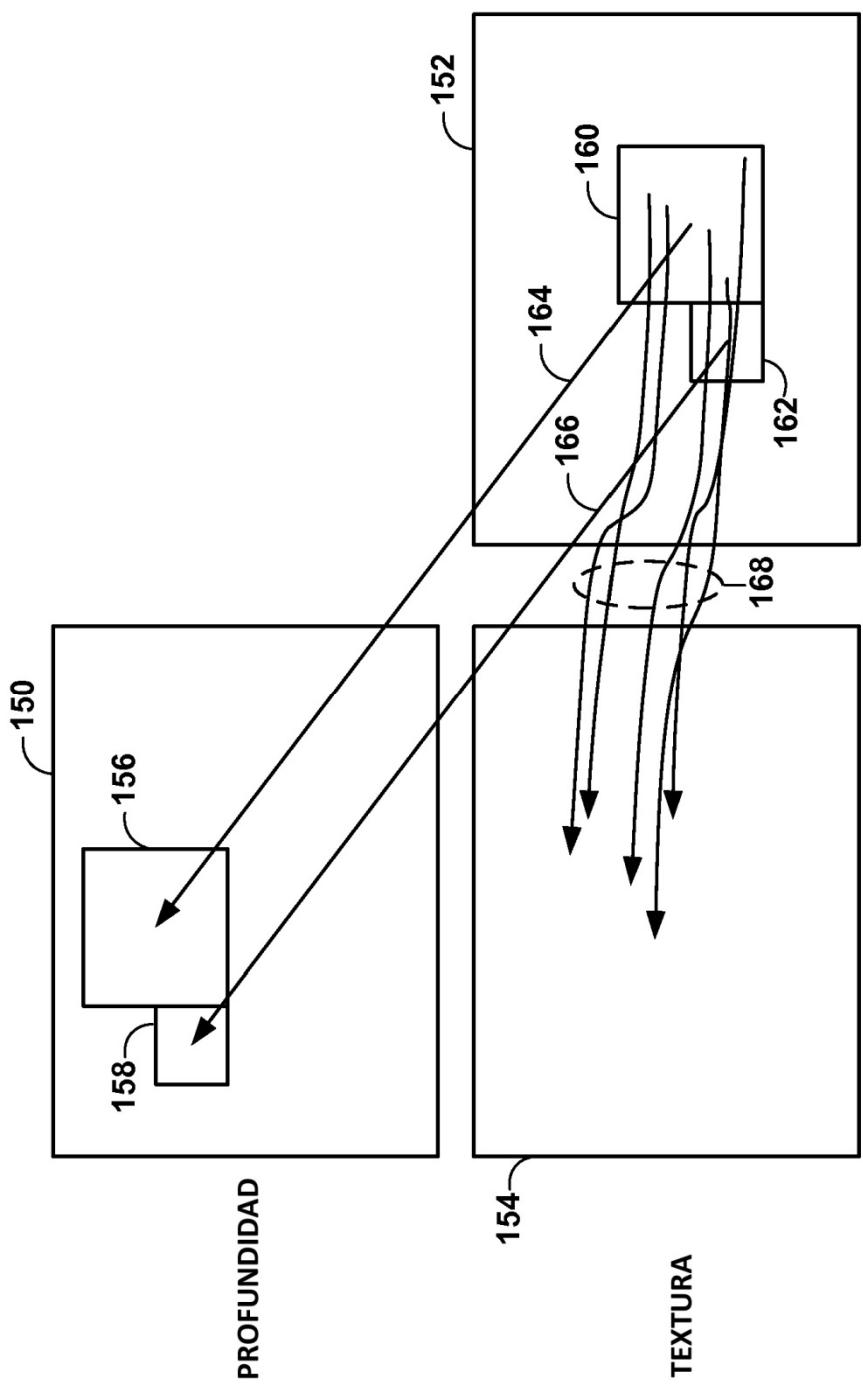


**FIG. 3**  
Una estructura de predicción entre vistas y temporal de MVC de ejemplo.

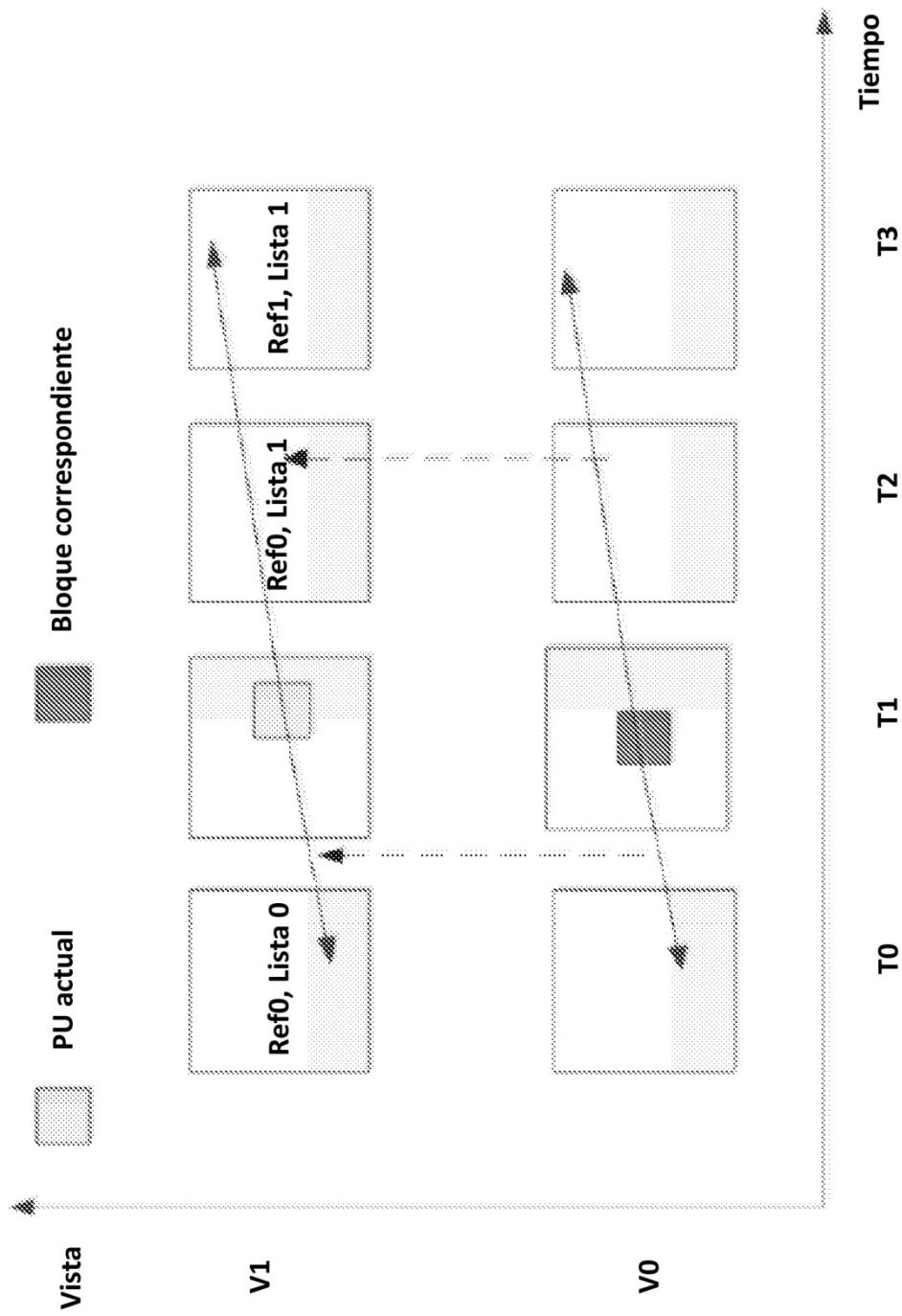
**FIG. 4**



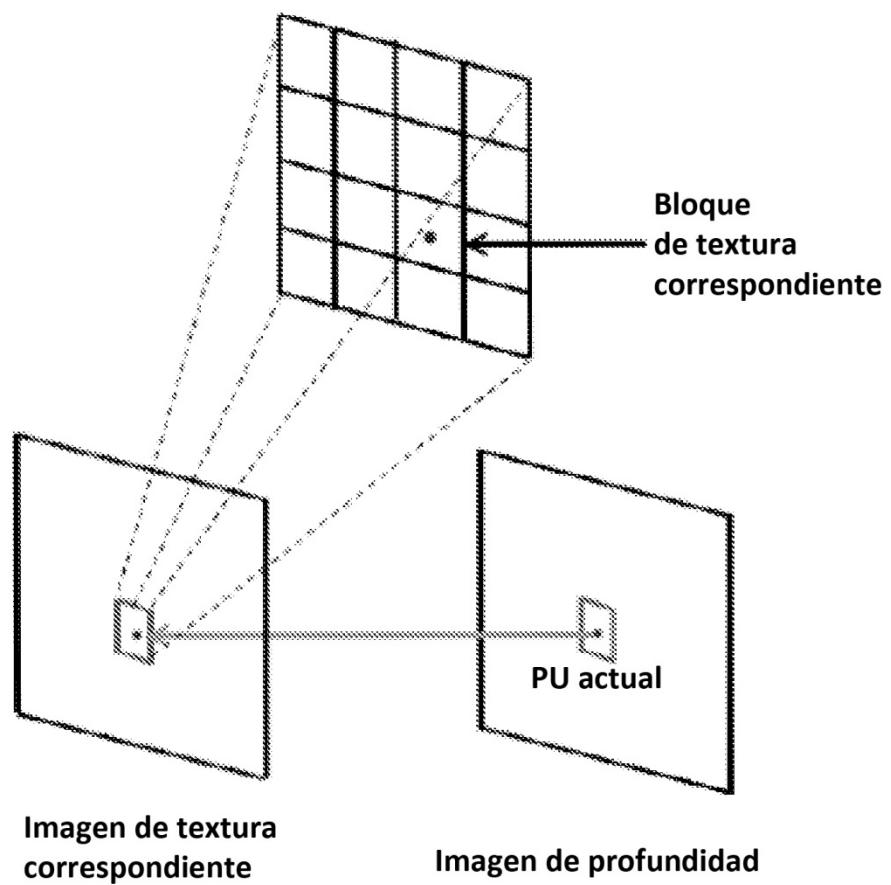
**FIG. 5**  
**Bloques vecinos temporales en NBDV.**



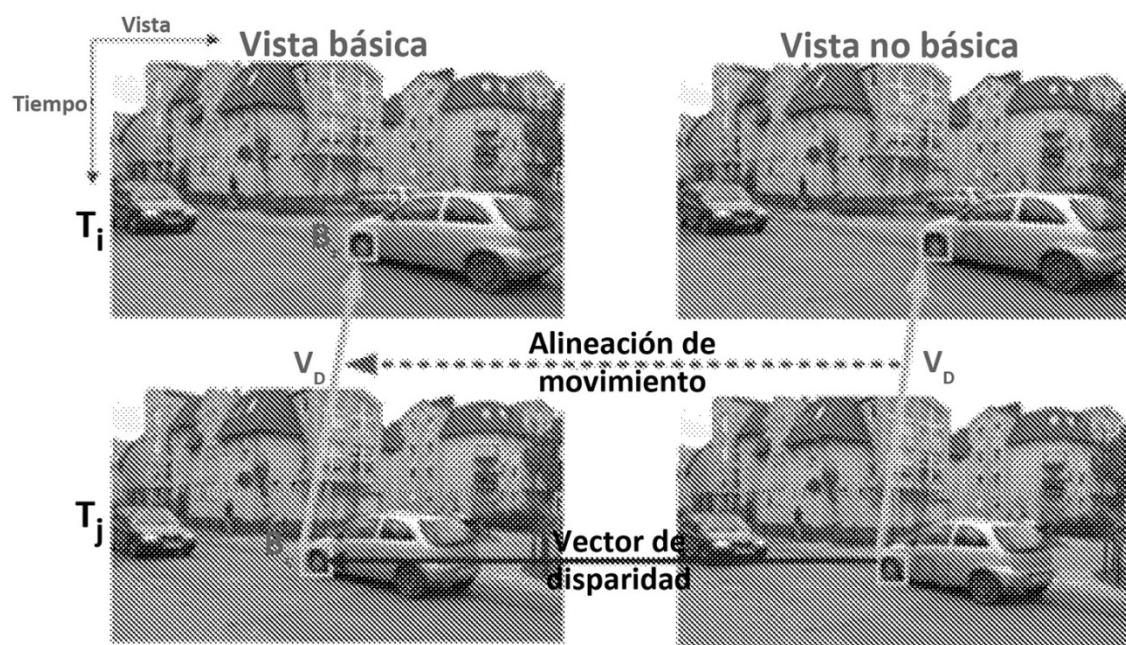
**FIG. 6**  
**Derivación del bloque de profundidad a partir de una vista de referencia para realizar una predicción BVSP**



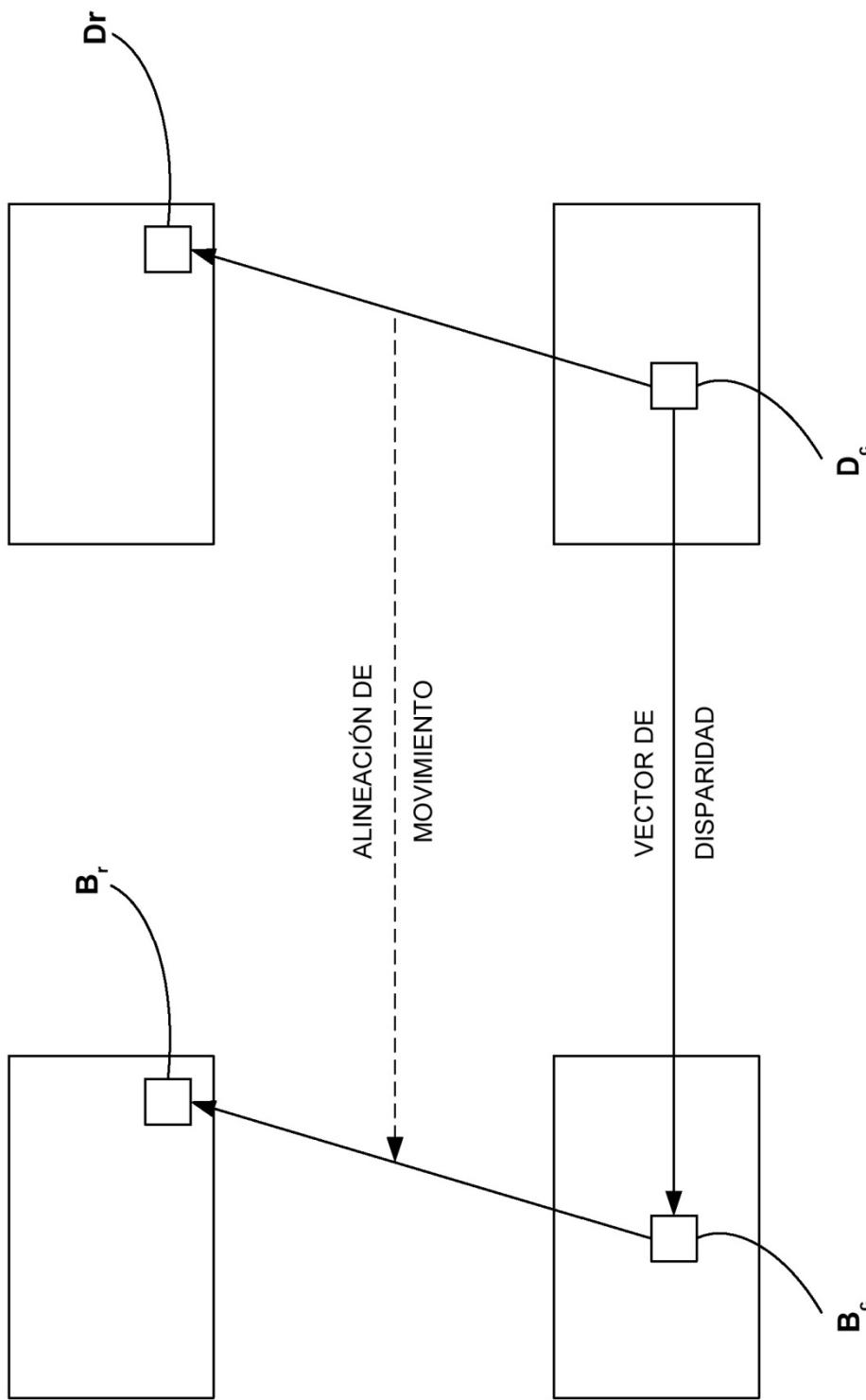
**FIG. 7**  
**Derivación del candidato de vector de movimiento predicho entre vistas para el modo de fusión/salto**



**FIG. 8**  
**Derivación del candidato MVI para**  
**codificación de profundidad**



**FIG. 9**  
**Estructura de predicción de la predicción residual avanzada.**



**FIG. 10**  
**Estructura de predicción de la predicción residual avanzada.**

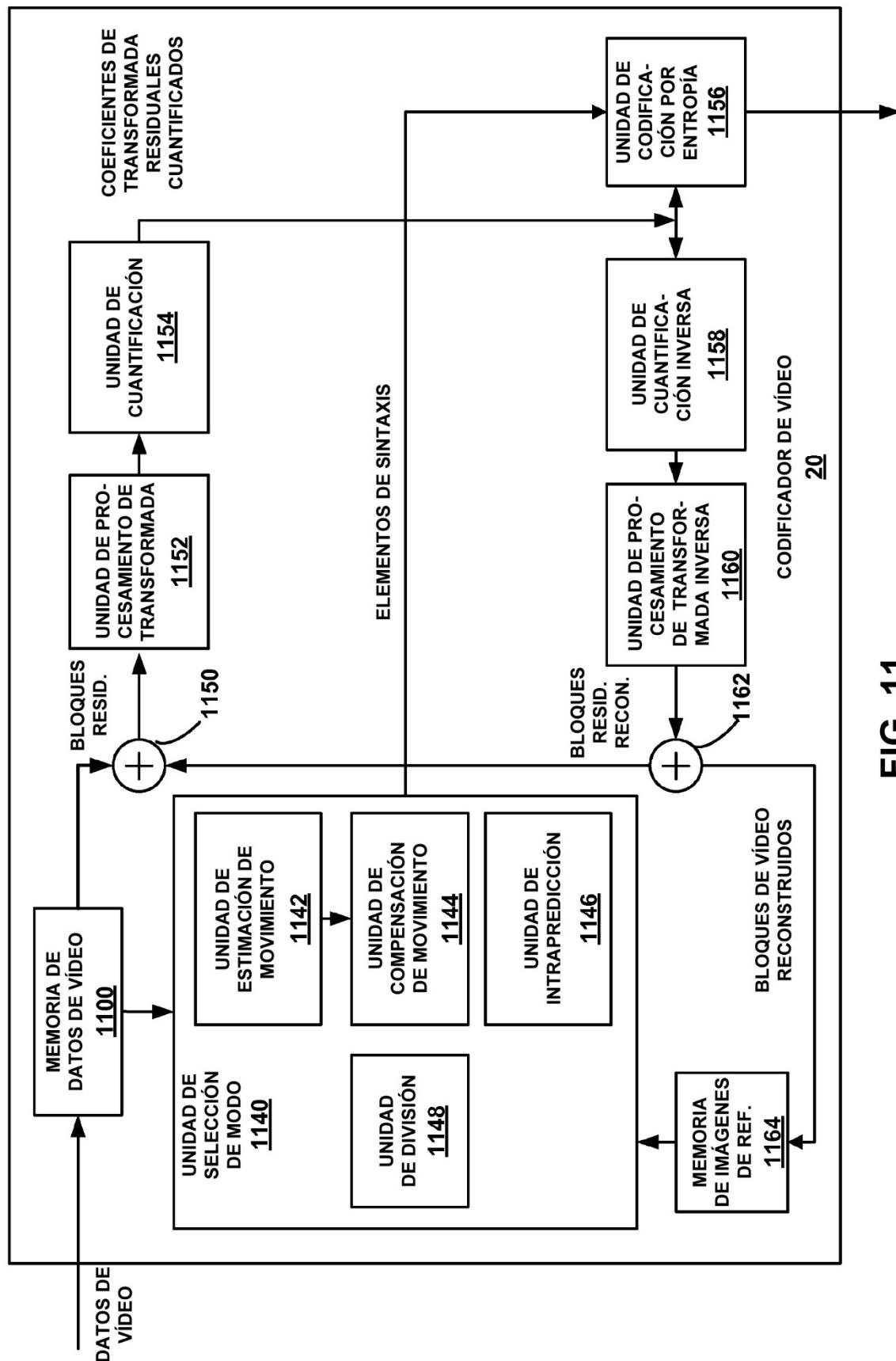


FIG. 11

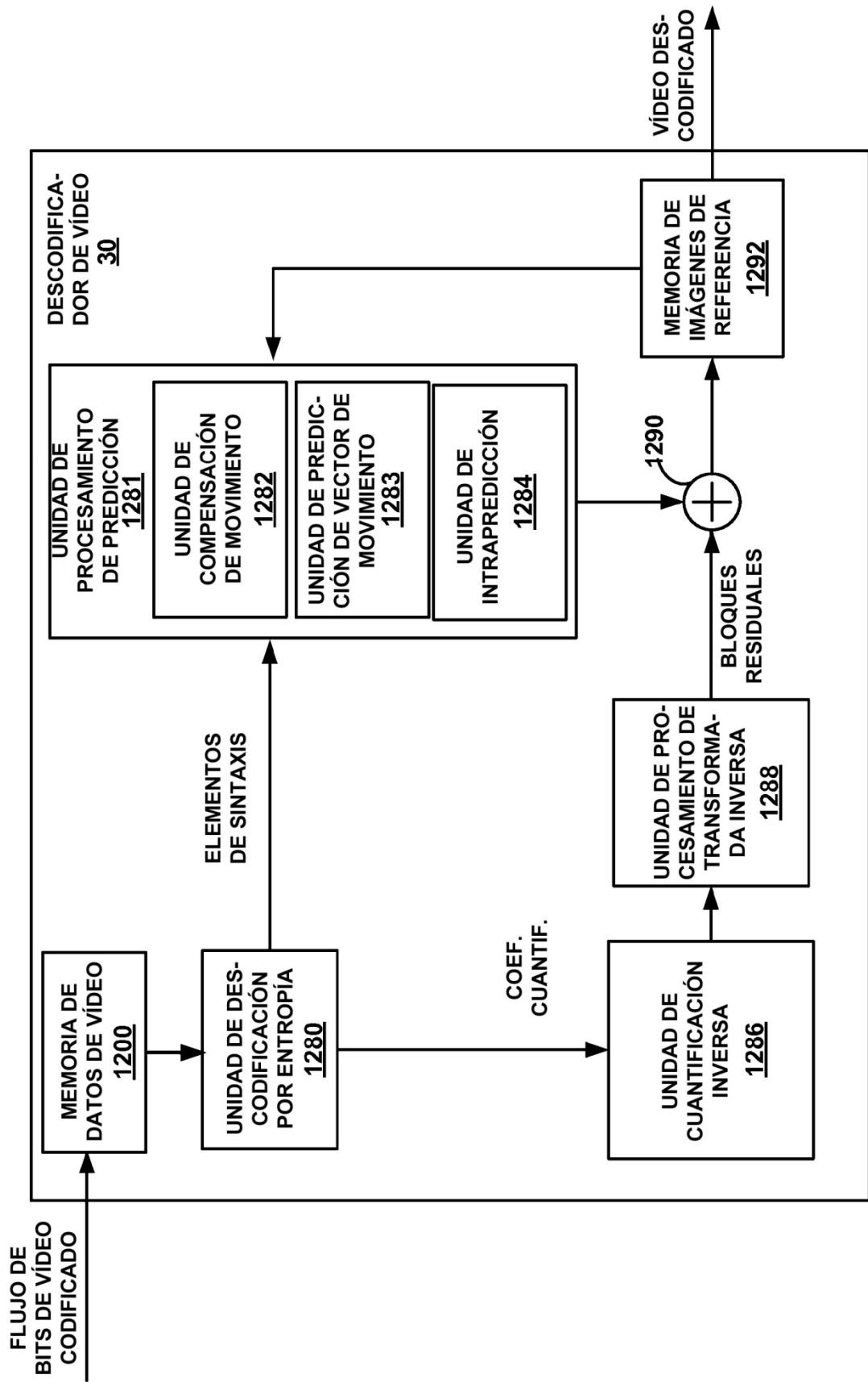
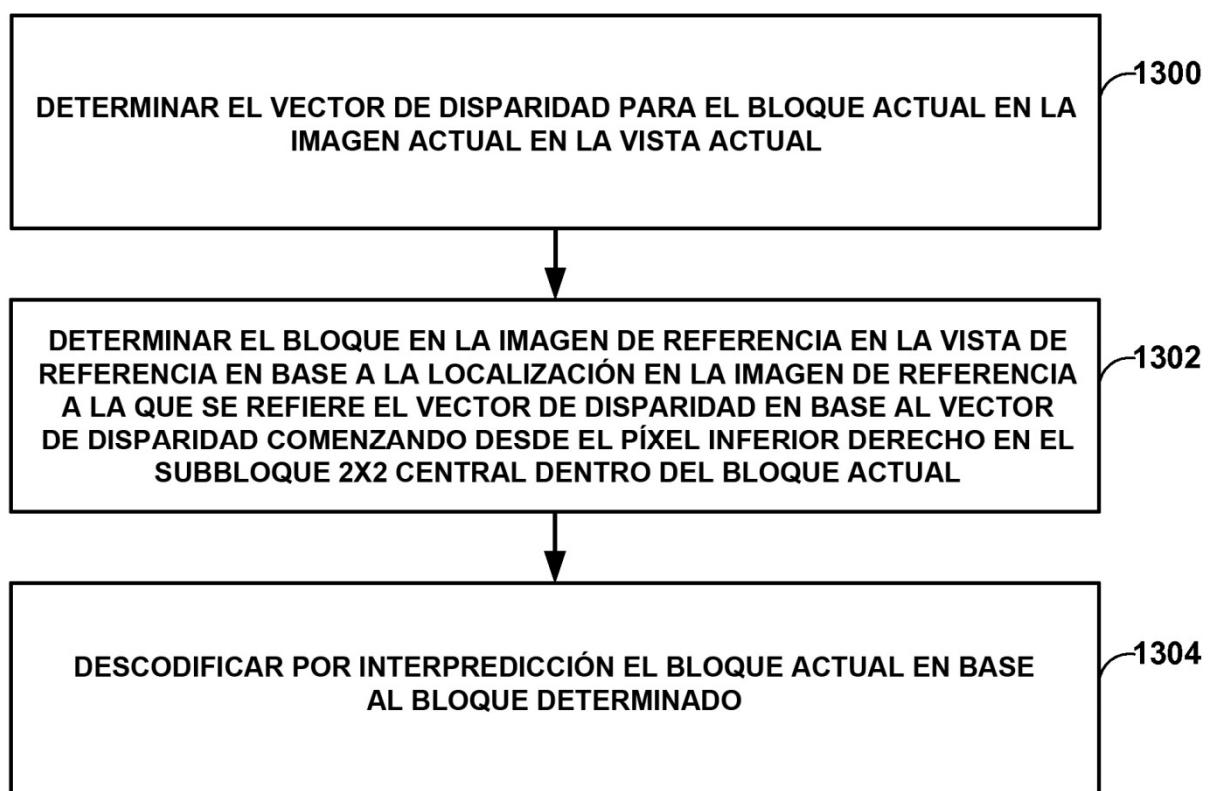
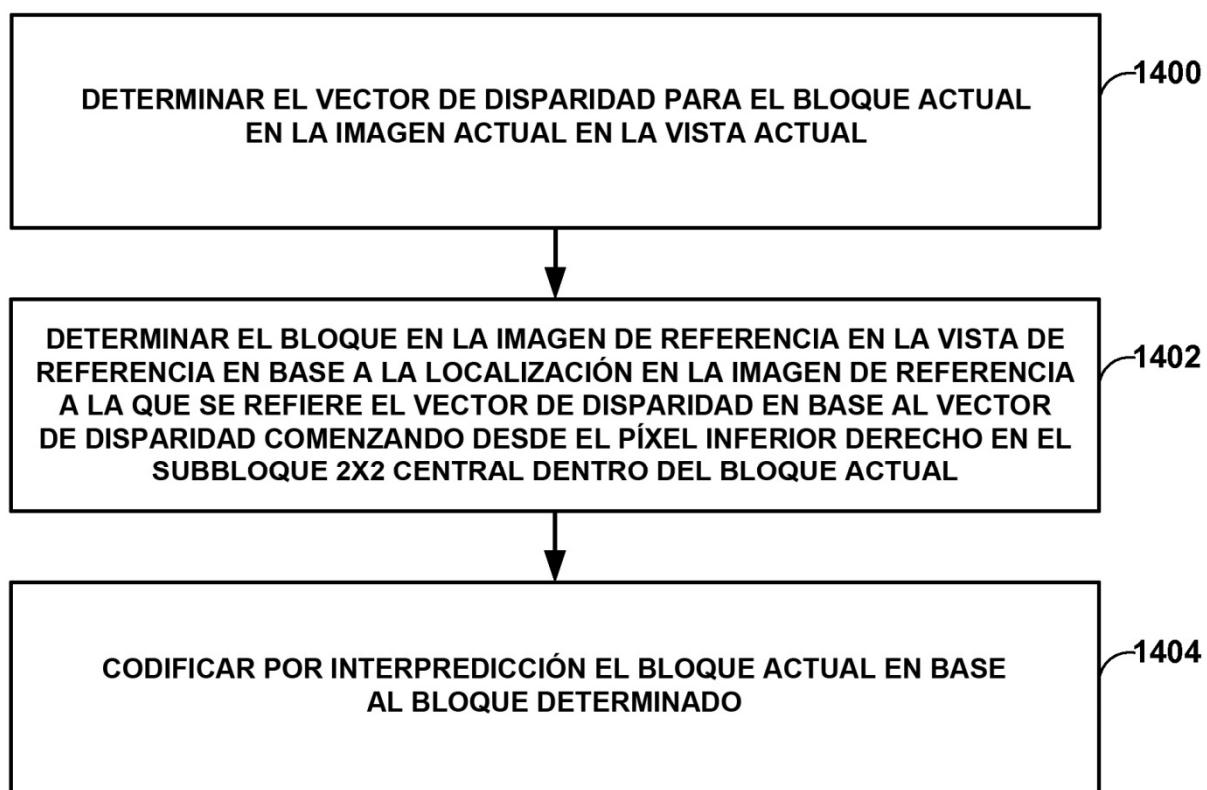


FIG. 12



**FIG. 13**



**FIG. 14**