(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 753 958**

(51) Int. Cl.:
H04N 19/00 (2014.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **26.09.2014 PCT/US2014/057732**

(87) Fecha y número de publicación internacional: **02.04.2015 WO15048453**

(96) Fecha de presentación y número de la solicitud europea: **26.09.2014 E 14782053 (4)**

(97) Fecha y número de publicación de la concesión europea: **07.08.2019 EP 3050295**

(54) Título: **Predicción de vector de movimiento temporal basada en subunidad de predicción (PU) en la HEVC y diseño de sub-PU en la 3D-HEVC**

(30) Prioridad:

26.09.2013 US 201361883111 P
07.10.2013 US 201361887915 P
17.10.2013 US 201361892327 P
31.10.2013 US 201361898385 P
25.09.2014 US 201414496991

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
15.04.2020

(73) Titular/es:

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

(72) Inventor/es:

CHEN, YING;
VAN DER AUWERA, GEERT;
ZHANG, LI y
LIU, HONGBIN

(74) Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 753 958 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

DESCRIPCIÓN

Predicción de vector de movimiento temporal basada en subunidad de predicción (PU) en la HEVC y diseño de sub-PU en la 3D-HEVC

CAMPO TÉCNICO

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

ANTECEDENTES

[0002] Las capacidades de vídeo digital pueden incorporarse a una amplia gama de dispositivos, que incluye televisores digitales, sistemas de radiodifusión digital directa, 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 celulares o de radio por satélite, los denominados «teléfonos inteligentes», dispositivos de videoconferencia, dispositivos de transmisión de vídeo en tiempo real y similares. Los dispositivos de vídeo digital implementan técnicas de compresión de vídeo, tales como las descritas en las normas definidas por MPEG-2, MPEG-4, ITU-T H.263, ITU-T TH.264/MPEG-4, Parte 10, Codificación de Vídeo Avanzada (AVC), la norma de Codificación de Vídeo de Alta Eficacia (HEVC) y las extensiones de tales normas. Los dispositivos de vídeo pueden transmitir, recibir, codificar, decodificar y/o almacenar información de vídeo digital con más eficacia implementando dichas técnicas de compresión de vídeo.

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

[0004] La predicción espacial o temporal da como resultado un bloque predictivo para un bloque a codificar. Los datos residuales representan diferencias de píxeles entre el bloque original a codificar y el bloque predictivo. Un bloque intercodificado se codifica de acuerdo a un vector de movimiento que apunta a un bloque de muestras de referencia que forman el bloque predictivo, y los datos residuales que indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intracodificado se codifica de acuerdo a una modalidad de intracodificación y a los datos residuales. Para una mayor compresión, los datos residuales pueden transformarse desde el dominio del píxel a un dominio de transformación, dando como resultado coeficientes de transformación residuales, que a continuación se pueden cuantizar. Los coeficientes de transformación cuantizados, dispuestos inicialmente en una formación bidimensional, pueden explorarse con el fin de generar un vector unidimensional de coeficientes de transformación, y puede aplicarse codificación por entropía para lograr aún más compresión. El documento US 2006/002474 describe un procedimiento para la estimación de movimiento de múltiples bloques. Solo se seleccionan unos pocos tamaños de bloque representativos para la estimación de movimiento cuando ocurren ciertas condiciones favorables. En el documento de Tian et al, "CEl.h: Backward View Synthesis Prediction using Neighboring Blocks", Equipo de Colaboración Conjunta en el Desarrollo de la Extensión de Codificación de Vídeo 3D de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, tercera conferencia: Ginebra, CH. del 16 al 23 de enero de 2013, número de documento JCT3V-00152, se propone un enfoque VSP de retrodeformación (BVSP). El enfoque BVSP propuesto usa los bloques vecinos para obtener un bloque de profundidad para realizar la operación de retrodeformación. Se propone la adición de un nuevo candidato de fusión que indique la modalidad BVSP en la lista de candidatos de fusión. En el documento de patente US 2006/0002474 se analiza un nuevo procedimiento, sistema y aparato para la estimación eficiente de movimiento multibloque en un esquema de compresión y codificación de señales digitales. La presente invención selecciona solamente unos pocos tamaños de bloque representativos para la estimación de movimiento cuando ocurren determinadas condiciones favorables, en lugar de usar todos los tamaños de bloque disponibles. En el documento de Sullivan et al, "Overview of the High Efficiency Video Coding (HEVC) Standard", TRANSACCIONES DEL IEEE SOBRE CIRCUITOS Y SISTEMAS PARA TECNOLOGÍA DE VÍDEO, VOL. 22, NO. 12, diciembre de 2012, se proporciona una visión general de los aspectos técnicos y características de la norma HEVC.

RESUMEN

[0005] En general, esta divulgación describe técnicas para la predicción de movimiento basada en la subunidad de predicción (PU), para la codificación de vídeo en la norma de codificación de vídeo de alta eficacia (HEVC) y en la extensión tridimensional de la norma HEVC. En un ejemplo, esta divulgación describe técnicas para realizar una modalidad avanzada de predicción de vectores de movimiento temporal (TMVP) para predecir las sub-PU de una PU

en la codificación de una sola capa para la cual se puede admitir el refinamiento del vector de movimiento. Convencionalmente, el diseño de sub-PU solo se habilita para la predicción de movimiento entre capas o entre vistas, utilizando una modalidad de interpredicción de fusión que no admite un refinamiento adicional de los vectores de movimiento predichos. La modalidad de TMVP avanzada incluye la determinación de vectores de movimiento para la PU en al menos dos etapas para obtener información de movimiento para la PU que incluye diferentes vectores de movimiento e índices de referencia para cada una de las sub-PU de la PU.

[0006] La extensión de la norma 3D-HEVC presta soporte a la codificación de vídeo de múltiples vistas más el formato de profundidad. En un ejemplo, en una modalidad de predicción de síntesis de retrovisión (BVSP) de sub-PU, la información de movimiento para una PU actual se predice a partir de un vector de movimiento de disparidad y una imagen asociada de referencia entre vistas, y el vector de movimiento de disparidad se refina en función de información de movimiento de sub-PU, asociada a bloques de profundidad de la imagen de referencia entre vistas. En la modalidad BVSP de sub-PU, la compensación de movimiento se realiza para predecir cada una de las sub-PU de la PU en función de la información independiente de movimiento de sub-PU. Convencionalmente, después de realizar la compensación de movimiento para predecir cada una de las sub-PU, solo se almacena el vector de movimiento de disparidad para la PU actual. De acuerdo a las técnicas de esta divulgación, para cada PU predicha utilizando la modalidad BVSP de sub-PU, un dispositivo de codificación de vídeo almacena la información de movimiento independiente, obtenida para cada una de las sub-PU de la PU actual, incluso después de que se realiza la compensación de movimiento. La información de movimiento adicional, almacenada para la PU actual, se puede usar luego para predecir las PU posteriores, para las cuales la PU actual es un bloque vecino.

[0007] Además, esta divulgación describe técnicas para aplicar un filtro de desbloqueo a cada unidad de codificación (CU) de un bloque de vídeo, para filtrar las fronteras de la unidad de transformación (TU) y las fronteras de la PU, incluidos las fronteras de sub-PU dentro de la CU, creando fronteras de PU artificiales o fronteras de TU artificiales en las fronteras de sub-PU.

[0008] En un ejemplo, esta divulgación está dirigida a un procedimiento de procesamiento de datos de vídeo que comprende determinar un vector de movimiento de primera etapa para una PU de una CU de un bloque de vídeo a partir de bloques vecinos de la PU como una primera etapa de una modalidad de TMVP avanzada, en el que el vector de movimiento de primera etapa identifica un bloque de una imagen de referencia correspondiente a la PU; dividir la PU en dos o más sub-PU; determinar información de movimiento de segunda etapa para cada una de las sub-PU del bloque de la imagen de referencia identificada por el vector de movimiento de primera etapa como una segunda etapa de la modalidad de TMVP avanzada, en el que la información de movimiento de segunda etapa para cada una de las sub-PU incluye al menos un vector de movimiento y un índice de referencia asociado; y realizar una compensación de movimiento para cada una de las sub-PU por separado basándose en la información de movimiento de segunda etapa para cada una de las sub-PU.

[0009] En otro ejemplo, esta divulgación está dirigida a un dispositivo de procesamiento de vídeo que comprende una memoria configurada para almacenar datos de vídeo; y a uno o más procesadores en comunicación con la memoria. El uno o más procesadores están configurados para determinar un vector de movimiento de primera etapa para una PU de una CU de un bloque de vídeo a partir de bloques vecinos de la PU como una primera etapa de una modalidad de TMVP avanzada, en el que el vector de movimiento de primera etapa identifica un bloque de una imagen de referencia correspondiente a la PU, dividir la PU en dos o más sub-PU, determinar información de movimiento de segunda etapa para cada una de las sub-PU del bloque de la imagen de referencia identificada por el vector de movimiento de primera etapa como una segunda etapa de la modalidad de TMVP avanzada, en el que la información de movimiento de segunda etapa para cada una de las sub-PU incluye al menos un vector de movimiento y un índice de referencia asociado, y realizar una compensación de movimiento para cada una de las sub-PU por separado basándose en la información de movimiento de segunda etapa para cada una de las sub-PU.

[0010] En un ejemplo adicional, esta divulgación está dirigida a un dispositivo de procesamiento de vídeo que comprende medios para determinar un vector de movimiento de primera etapa para una PU de una CU de un bloque de vídeo a partir de bloques vecinos de la PU como una primera etapa de una modalidad de TMVP avanzada, en el que el vector de movimiento de primera etapa identifica un bloque de una imagen de referencia correspondiente a la PU; medios para dividir la PU en dos o más sub-PU; medios para determinar información de movimiento de segunda etapa para cada una de las sub-PU del bloque de la imagen de referencia identificada por el vector de movimiento de primera etapa como una segunda etapa de la modalidad de TMVP avanzada, en el que la información de movimiento de segunda etapa para cada una de las sub-PU incluye al menos un vector de movimiento y un índice de referencia asociado; y medios para realizar una compensación de movimiento para cada una de las sub-PU por separado basándose en la información de movimiento de segunda etapa para cada una de las sub-PU.

[0011] En un ejemplo adicional, esta divulgación está dirigida a un medio de almacenamiento legible por ordenador que almacena instrucciones para procesar datos de vídeo que, cuando se ejecutan, hacen que uno o más procesadores determinen un vector de movimiento de primera etapa para una PU de una CU de un bloque de vídeo a partir de bloques vecinos de la PU como una primera etapa de una modalidad de TMVP avanzada, en el que el vector de movimiento de primera etapa identifica un bloque de una imagen de referencia correspondiente a la PU; dividan la PU en dos o más sub-PU; determinen información de movimiento de segunda etapa para cada una de las

sub-PU del bloque de la imagen de referencia identificada por el vector de movimiento de primera etapa como una segunda etapa de la modalidad de TMVP avanzada, en el que la información de movimiento de segunda etapa para cada una de las sub-PU incluye al menos un vector de movimiento y un índice de referencia asociado; y realicen una compensación de movimiento para cada una de las sub-PU por separado basándose en la información de movimiento de segunda etapa para cada una de las sub-PU.

[0012] En otro ejemplo, esta divulgación está dirigida a un procedimiento de procesamiento de datos de vídeo, donde el procedimiento comprende determinar primera información de movimiento para una PU de una CU de un bloque de vídeo a partir de bloques vecinos de la PU de acuerdo con una modalidad de BVSP, en el que la primera información de movimiento incluye al menos un vector de movimiento de disparidad y un índice de vista de referencia asociado que identifica una imagen de referencia entre vistas; dividir la PU en dos o más sub-PU; determinar segunda información de movimiento para cada una de las sub-PU, en el que la segunda información de movimiento incluye al menos un vector de movimiento de disparidad asociado a un bloque de profundidad de la imagen de referencia entre vistas correspondiente a cada una de las sub-PU; realizar una compensación de movimiento para predecir cada una de las sub-PU a partir de la imagen de referencia entre vistas basándose en la segunda información de movimiento; y almacenar la segunda información de movimiento para cada una de las sub-PU de la PU en una memoria, que se va a usar para predecir PU posteriores.

[0013] En otro ejemplo, esta divulgación está dirigida a un dispositivo de procesamiento de vídeo que comprende una memoria configurada para almacenar datos de vídeo, y a uno o más procesadores en comunicación con la memoria. El uno o más procesadores están configurados para determinar primera información de movimiento para una PU de una CU de un bloque de vídeo a partir de bloques vecinos de la PU de acuerdo con una modalidad de BVSP, en el que la primera información de movimiento incluye al menos un vector de movimiento de disparidad y un índice de vista de referencia asociado que identifica una imagen de referencia entre vistas, dividir la PU en dos o más sub-PU, determinar segunda información de movimiento para cada una de las sub-PU, en el que la segunda información de movimiento incluye al menos un vector de movimiento de disparidad asociado a un bloque de profundidad de la imagen de referencia entre vistas correspondiente a cada una de las sub-PU, y realizar una compensación de movimiento para predecir cada una de las sub-PU a partir de la imagen de referencia entre vistas basándose en la segunda información de movimiento. La memoria está configurada para almacenar la segunda información de movimiento para cada una de las sub-PU de la PU, que se va a usar para predecir PU posteriores.

[0014] En un ejemplo adicional, esta divulgación está dirigida a un dispositivo de procesamiento de vídeo que comprende medios para determinar primera información de movimiento para una PU de una CU de un bloque de vídeo a partir de bloques vecinos de la PU de acuerdo con una modalidad de BVSP, en el que la primera información de movimiento incluye al menos un vector de movimiento de disparidad y un índice de vista de referencia asociado que identifica una imagen de referencia entre vistas; medios para dividir la PU en dos o más sub-PU; medios para determinar segunda información de movimiento para cada una de las sub-PU, en el que la segunda información de movimiento incluye al menos un vector de movimiento de disparidad asociado a un bloque de profundidad de la imagen de referencia entre vistas correspondiente a cada una de las sub-PU; medios para realizar una compensación de movimiento para predecir cada una de las sub-PU a partir de la imagen de referencia entre vistas basándose en la segunda información de movimiento; y medios para almacenar la segunda información de movimiento para cada una de las sub-PU de la PU en una memoria, que se va a usar para predecir PU posteriores.

[0015] En otro ejemplo, esta divulgación está dirigida a un medio de almacenamiento legible por ordenador que almacena instrucciones para procesar datos de vídeo que, cuando se ejecutan, hacen que uno o más procesadores determinen primera información de movimiento para una PU de una CU de un bloque de vídeo a partir de bloques vecinos de la PU de acuerdo con una modalidad de BVSP, en el que la primera información de movimiento incluye al menos un vector de movimiento de disparidad y un índice de vista de referencia asociado que identifica una imagen de referencia entre vistas; dividan la PU en dos o más sub-PU; determinen segunda información de movimiento para cada una de las sub-PU, en el que la segunda información de movimiento incluye al menos un vector de movimiento de disparidad asociado a un bloque de profundidad de la imagen de referencia entre vistas correspondiente a cada una de las sub-PU; realicen una compensación de movimiento para predecir cada una de las sub-PU a partir de la imagen de referencia entre vistas basándose en la segunda información de movimiento; y almacenar la segunda información de movimiento para cada una de las sub-PU de la PU en una memoria, que se va a usar para predecir PU posteriores.

[0016] Los detalles de uno o más ejemplos se exponen en los dibujos adjuntos y en la descripción siguiente. Otras características, objetos y ventajas resultarán evidentes a partir de la descripción y de los dibujos, y a partir de las reivindicaciones. La invención se expone en las reivindicaciones adjuntas.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

[0017]

La figura 1 es un diagrama de bloques que ilustra un sistema ejemplar de codificación y decodificación de vídeo que puede utilizar las técnicas de la presente divulgación.

La figura 2 es un diagrama conceptual que ilustra ocho modalidades de partición para definir unidades de predicción (PU) de una unidad de codificación (CU).

La figura 3 es un diagrama conceptual que ilustra una estructura de árbol de transformación dentro de una CU.

Las figuras 4A y 4B son diagramas conceptuales que ilustran, respectivamente, candidatos adyacentes espaciales para la modalidad de fusión y la modalidad de predicción de vectores de movimiento (AMVP) avanzada.

La figura 5A es un diagrama conceptual que ilustra los candidatos de predicción de vectores de movimiento temporal (TMVP).

La figura 5B es un diagrama conceptual que ilustra el ajuste a escala del vector de movimiento (MV) para la TMVP.

La figura 6 es un diagrama de flujo que ilustra un flujo de procesamiento ejemplar de un proceso de filtro de desbloqueo.

La figura 7 es un diagrama de flujo que ilustra una operación ejemplar de un cálculo de resistencia de frontera (Bs) para un proceso de filtro de desbloqueo.

La figura 8 es un diagrama conceptual que ilustra información reutilizada o referida para un cálculo de Bs en una frontera de unidad de árbol de codificación (CTU) para un proceso de filtro de desbloqueo.

La figura 9 es un diagrama conceptual que ilustra los píxeles implicados en una decisión de filtro de encendido/apagado y una selección de filtro fuerte/débil para un proceso de filtro de desbloqueo.

La figura 10 es un diagrama conceptual que ilustra una estructura de predicción de múltiples vistas para casos de 3 vistas, donde V3 indica la vista base y una imagen en una vista no base (es decir, V1 o V5) puede predecirse a partir de imágenes en la vista base de la misma instancia temporal.

La figura 11 es un diagrama conceptual que ilustra los bloques vecinos espaciales a los que se accede para la obtención del vector de disparidad basado en bloques vecinos (NBDV), donde los bloques vecinos espaciales para la obtención del NBDV son los mismos a los que se accede para la modalidad de fusión y la modalidad de AMVP.

La figura 12 es un diagrama conceptual que ilustra una modalidad de predicción de síntesis de retrovisión (BVSP) que asigna diferentes vectores de movimiento para diferentes sub-PU de una PU.

La figura 13 es un diagrama conceptual que ilustra la predicción de movimiento entre vistas al nivel de sub-PU.

Las figuras 14A y 14B son diagramas conceptuales que ilustran una modalidad de TMVP avanzada para predecir las sub-PU en una PU en codificación de capa única.

La figura 15 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo que puede implementar las técnicas de esta divulgación.

La figura 16 es un diagrama de bloques que ilustra un ejemplo de un decodificador de vídeo que puede implementar las técnicas de esta divulgación.

La figura 17 es un diagrama de flujo que ilustra una operación ejemplar de predicción de una PU actual utilizando una modalidad de BVSP de sub-PU, y de almacenamiento de la información de movimiento de sub-PU determinada.

La figura 18 es un diagrama de flujo que ilustra una operación ejemplar de aplicación de un filtro de desbloqueo a cada CU de un bloque de vídeo para filtrar las fronteras de TU y las fronteras de PU, incluidas las fronteras de sub-PU dentro de la CU.

La figura 19 es un diagrama de flujo que ilustra una operación ejemplar de una modalidad de TMVP avanzada para predecir las sub-PU de una PU en codificación de capa única.

DESCRIPCIÓN DETALLADA

[0018] Esta divulgación describe técnicas para la predicción de movimiento a nivel de subunidad de predicción (PU), para la codificación de vídeo usando la extensión tridimensional a la norma de codificación de vídeo de alta eficacia (HEVC). La extensión de la norma 3D-HEVC presta soporte a la codificación de vídeo de múltiples vistas más el formato de profundidad. En la codificación de vídeo de múltiples vistas, una imagen de vídeo actual se puede predecir, tanto con imágenes de referencia temporales en una misma vista, como con imágenes de referencia entre vistas en

una vista de referencia.

[0019] En un ejemplo, la predicción de síntesis de vista en la norma 3D-HEVC se puede realizar utilizando una modalidad de predicción de síntesis de retrovisión (BVSP) de sub-PU. En la modalidad de BVSP de sub-PU, la información de movimiento para una PU actual se predice a partir de un vector de movimiento de disparidad y un índice asociado de vista de referencia, que identifica una imagen de referencia entre vistas. En la modalidad de BVSP de sub-PU, la PU actual se divide en dos o más sub-PU, y el vector de movimiento de disparidad se refina en función de la información de movimiento de sub-PU obtenida para cada una de las sub-PU en función de un correspondiente bloque de profundidad de la imagen de referencia entre vistas. La compensación de movimiento se realiza para predecir cada una de las sub-PU de la PU en función de la información de movimiento de sub-PU independiente. Convencionalmente, después de realizar la compensación de movimiento para predecir cada una de las sub-PU, solo se almacena el vector de movimiento de disparidad para la PU actual.

[0020] De acuerdo a las técnicas de esta divulgación, para cada PU predicha utilizando la modalidad BVSP de sub-PU, un dispositivo de codificación de vídeo almacena la información de movimiento independiente, obtenida para cada una de las sub-PU de la PU actual, incluso después de que se realiza la compensación de movimiento. La información de movimiento adicional, almacenada para la PU actual, se puede usar luego para predecir las PU posteriores, para las cuales la PU actual es un bloque vecino.

[0021] Como otro ejemplo, esta divulgación describe técnicas para realizar una modalidad avanzado de predicción de vectores de movimiento temporal (TMVP), para predecir las sub-PU de una PU en la codificación de una sola capa, para la cual se pueda admitir el refinamiento del vector de movimiento. Convencionalmente, el diseño de sub-PU solo se habilita para la predicción de movimiento entre capas o entre vistas, utilizando una modalidad de interpredicción de fusión que no admite un refinamiento adicional de los vectores de movimiento predichos. La modalidad de TMVP avanzada incluye la determinación de vectores de movimiento para la PU en al menos dos etapas para obtener información de movimiento para la PU que incluye diferentes vectores de movimiento e índices de referencia para cada una de las sub-PU de la PU. Una sub-PU dentro de una PU puede ser de tamaño NxN o NxM, en donde N y M pueden ser números enteros cualesquiera. Los tamaños típicos de sub-PU pueden ser 4x4, 8x8, 4x8 u 8x4.

[0022] De acuerdo a las técnicas de esta divulgación, se determina un vector de movimiento de primera etapa para una PU actual a partir de bloques vecinos de la PU, donde el vector de movimiento de primera etapa identifica un bloque de una imagen de referencia correspondiente a la PU actual. De manera similar a la predicción de movimiento de sub-PU en la norma 3D-HEVC, el tamaño del área correspondiente (por ejemplo, el bloque) de la imagen de referencia que incluye las sub-PU más pequeñas es el mismo que el tamaño de la PU actual. De esta manera, las sub-PU de la PU actual tienen las sub-PU correspondientes en el área correspondiente de la imagen de referencia identificada con un vector de movimiento único. El vector de movimiento de la primera etapa puede identificarse basándose en vecinos espaciales y/o temporales. Alternativamente, el vector de movimiento de la primera etapa se puede configurar para que sea constante, por ejemplo, 0 u otro valor predefinido, y se relaciona con el tamaño de la PU actual. En otra alternativa, cada una de las sub-PU de la PU puede identificar su bloque correspondiente de la imagen de referencia con un vector de movimiento diferente.

[0023] Como se ha descrito anteriormente, la PU actual se divide en dos o más sub-PU, y la información de movimiento de la segunda etapa se determina para cada una de las sub-PU desde el área correspondiente, por ejemplo, el bloque, de la imagen de referencia identificada por el vector de movimiento de la primera etapa, donde la información de movimiento de la segunda etapa, para cada una de las sub-PU, incluye al menos un vector de movimiento y un índice de referencia asociado. La compensación de movimiento se realiza para predecir cada una de las sub-PU en función de la información de movimiento de la segunda etapa para cada una de las sub-PU. Este proceso avanzado de TMVP se puede lograr activando un candidato de fusión entre todos los demás candidatos de fusión, lo que significa que se crea un candidato de fusión específico para indicar que dicho proceso debe realizarse para la PU actual. En algunos ejemplos, en lugar de operar únicamente en la modalidad de fusión, se puede determinar una diferencia de vectores de movimiento para refinar el al menos un vector de movimiento de la información de movimiento de la segunda etapa para cada una de las sub-PU.

[0024] Como un ejemplo adicional, esta divulgación describe técnicas relacionadas con los procesos de filtro de desbloqueo, aplicados a unidades de codificación (CU) de un bloque de vídeo que incluyen al menos una PU con múltiples sub-PU. Los filtros de desbloqueo para la norma HEVC no están diseñados para filtrar dentro de una PU, es decir, a lo largo de fronteras de sub-PU entre sub-PU adyacentes de la PU, porque para los bloques de la HEVC se supone que la compensación de movimiento es la misma para toda la PU. Las técnicas de esta divulgación permiten que las fronteras de sub-PU se desbloqueen convirtiendo las sub-PU en una estructura favorecedora del desbloqueo, de modo que los filtros de desbloqueo de la HEVC puedan seguir utilizándose para los bloques de la norma 3D-HEVC. Las técnicas de desbloqueo propuestas pueden ser aplicables a códecs de múltiples capas, incluyendo la 3D-HEVC.

[0025] De acuerdo a las técnicas de esta divulgación, antes de aplicar un filtro de desbloqueo a una CU del bloque de vídeo que incluye la PU con las sub-PU, la CU se convierte para crear fronteras de PU artificiales o fronteras de TU artificiales en las fronteras de sub-PU. En un ejemplo, un árbol de transformación de la CU se convierte para asociar la PU con una jerarquía de árbol de transformación y asociar cada una de las sub-PU a una TU de modo que

las fronteras de sub-PU se conviertan en fronteras de TU artificiales. En otro ejemplo, la CU se convierte en un árbol de codificación para asociar la PU con una CU y asociar cada una de las sub-PU a una PU de tal manera que las fronteras de sub-PU se conviertan en fronteras de PU artificiales. El filtro de desbloqueo se puede aplicar luego a las fronteras de PU entre dos PU adyacentes de la CU y/o las fronteras de TU entre dos TU adyacentes de la CU, incluidas las fronteras de PU artificiales y las fronteras de TU artificiales.

[0026] Entre las normas de codificación de vídeo se incluyen ITU-T H.261, MPEG-1 Visual de la ISO/IEC, ITU-T H.262 o MPEG-2 Visual de la ISO/IEC, ITU-T TH.263, MPEG-4 Visual de la ISO/IEC e ITU-T H.264 (también conocida como AVC del MPEG-4 de la ISO/IEC), incluyendo sus ampliaciones de codificación de vídeo ajustable a escala (SVC) y de codificación de vídeo de múltiples vistas (MVC). El más reciente borrador conjunto de la MVC se describe en ITU-TH.264, Serie H: Sistemas audiovisuales y de multimedios, Infraestructura de servicios audiovisuales - Codificación de vídeo en movimiento, Codificación de vídeo avanzada para servicios audiovisuales genéricos, Sector de normalización de las telecomunicaciones de la Unión Internacional de Telecomunicaciones (ITU), marzo de 2010.

[0027] Además, el diseño de una nueva norma de codificación de vídeo, concretamente la Codificación de Vídeo de Alta Eficacia (HEVC), ha sido finalizado 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 del Grupo de Expertos en Imágenes en Movimiento (MPEG) de la ISO/IEC. Un borrador de la norma HEVC denominado "Borrador de trabajo 10 de la HEVC" o "WD10", descrito en el artículo de Bross et al., "High Efficiency vídeo Coding (HEVC) Text Specification Draft 10 (for FDIS & Last Call)" ["Memoria descriptiva textual de la Codificación de Vídeo de Alta Eficacia (HEVC), Borrador 10 (para FDIS y Última Llamada)"], Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 12ª conferencia: Ginebra, CH, 14 a 23 de enero de 2013, JCTVC-L1003v34, está disponible en http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v34.zip.

[0028] El documento estándar finalizado se publica como ITU-T H.265, Serie H: Sistemas audiovisuales y de multimedios, Infraestructura de servicios audiovisuales - Codificación de vídeo en movimiento, Codificación de vídeo de alta eficacia, Sector de normalización de las telecomunicaciones de la Unión Internacional de Telecomunicaciones (ITU), abril de 2013. Una reciente descripción del codificador de la HEVC, descrita en Il-Koo Kim et al., "Descripción del Codificador del Modelo de Prueba 10 (HM10) de la Codificación de vídeo de alta eficacia (HEVC), Equipo de colaboración conjunta sobre codificación de vídeo (JCT-VC) del UIT-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 12ª Conferencia: Ginebra, CH, 14 a 23 de enero de 2013, está disponible en: http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1002-v3.zip.

[0029] Dos extensiones de la HEVC con soporte de servicios tridimensionales han estado en desarrollo por parte del Equipo Conjunto de Colaboración en Codificación de Vídeo Tridimensional (JCT-3V) del Grupo de Expertos en Codificación de Vídeo (VCEG) del ITU-T y el Grupo de Expertos en Imágenes en Movimiento (MPEG) de la ISO/IEC. Las dos extensiones son la extensión de múltiples vistas para la HEVC (MV-HEVC) y la extensión tridimensional para la HEVC (3D-HEVC).

[0030] La MV-HEVC presta soporte a la codificación de múltiples vistas de textura sin cambiar el diseño a nivel de bloque de la HEVC. Un borrador de memoria descriptiva de la MV-HEVC, denominado Borrador de Trabajo 5 (WD5) de la MV-HEVC y descrito en el artículo de Tech et al., "Texto en borrador 5 de la MV-HEVC", Equipo de colaboración conjunta sobre desarrollo de extensión de codificación de vídeo tridimensional (JCT-3V) del ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 5ª Conferencia: Viena, AT, 27 de julio a 2 de agosto de 2013, JCT3V-E1004v6, está disponible en http://phenix.it-sudparis.eu/jct2/doc_end_user/documents/5_Vienna/wg11/JCT3V-E1004-v6.zip.

[0031] La norma 3D-HEVC presta soporte a la codificación de vídeo de múltiples vistas más el formato de profundidad e incluye nuevas herramientas de codificación integradas además de los módulos de codificación de la HEVC. Las herramientas de codificación recién introducidas son aplicables tanto para la codificación de texturas como para la codificación de profundidad. Un borrador de especificación de la norma 3D-HEVC, denominado Borrador de Trabajo 1 (WD1) de la norma 3D-HEVC y descrito en el artículo de Tech et al., "3D-HEVC Draft Text 1 [Texto en borrador 1 de 3D-HEVC]", Equipo de colaboración conjunta sobre desarrollo de extensión de codificación de vídeo tridimensional (JCT-3V) del ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 5ª Conferencia: Viena, AT, 27 de julio a 2 de agosto de 2013, JCT3V-E1001v3, está disponible en http://phenix.it-sudparis.eu/jct2/doc_end_user/documents/5_Viennatwg11/JCT3V-E1001-v3.zip.

[0032] La figura 1 es un diagrama de bloques que ilustra un sistema ejemplar de codificación y decodificación de vídeo 10 que puede utilizar técnicas de esta divulgación. Como se muestra en la figura 1, el sistema 10 incluye un dispositivo de origen 12 que proporciona datos de vídeo codificados, a decodificar más adelante por parte de un dispositivo de destino 14. En particular, el dispositivo de origen 12 proporciona los datos de vídeo al dispositivo de destino 14 mediante un medio legible por ordenador 16. El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender cualquiera entre una amplia gama de dispositivos, incluidos ordenadores de escritorio, ordenadores plegables (es decir, portátiles), ordenadores de tableta, decodificadores, equipos manuales telefónicos tales como los denominados teléfonos «inteligentes», los denominados paneles «inteligentes», televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de transmisión de vídeo en tiempo real o similares. En algunos casos, el dispositivo de origen 12 y el dispositivo de destino 14 pueden

estar equipados para la comunicación inalámbrica.

[0033] El dispositivo de destino 14 puede recibir los datos de vídeo codificados, a decodificar mediante el medio legible por ordenador 16. El medio legible por ordenador 16 puede comprender cualquier tipo de medio o dispositivo capaz de desplazar los datos de vídeo codificados desde el dispositivo de origen 12 al dispositivo de destino 14. En un ejemplo, el medio legible por ordenador 16 puede comprender un medio de comunicación que habilita el dispositivo de origen 12 para transmitir datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real. Los datos de vídeo codificados se pueden modular de acuerdo a una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitir al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación, inalámbrico o cableado, 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, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo de origen 12 hasta el dispositivo de destino 14.

[0034] En algunos ejemplos, pueden emitirse datos codificados desde la interfaz de salida 22 hasta un dispositivo de almacenamiento. De forma similar, se puede acceder a los datos codificados desde el dispositivo de almacenamiento mediante una interfaz de entrada. El dispositivo de almacenamiento puede incluir cualquiera entre varios medios de almacenamiento de datos, distribuidos o de acceso local, tales como un disco duro, discos Blu-ray, DVD, CD-ROM, memoria flash, memoria volátil o no volátil o cualquier otro medio de almacenamiento digital adecuado para almacenar datos de vídeo codificados. En otro ejemplo, el dispositivo de almacenamiento puede corresponder a un servidor de ficheros o a otro dispositivo de almacenamiento intermedio que pueda almacenar el vídeo codificado generado por el dispositivo de origen 12. El dispositivo de destino 14 puede acceder a datos de vídeo almacenados desde el dispositivo de almacenamiento, mediante transmisión continua o descarga. El servidor de archivos puede ser cualquier tipo de servidor capaz de almacenar datos de vídeo codificados y transmitir esos datos de vídeo codificados al dispositivo de destino 14. Los servidores ejemplares de ficheros incluyen un servidor de la Red (por ejemplo, para una sede de la Red), un servidor del FTP, dispositivos de almacenamiento conectados en red (NAS) o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de cualquier conexión de datos estándar, incluida una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión de Wi-Fi), una conexión cableada (por ejemplo, DSL, módem de cable, etc.) o una combinación de ambos que sea adecuada para acceder a datos de vídeo codificados almacenados en un servidor de ficheros. La transmisión de datos de vídeo codificados desde el dispositivo de almacenamiento puede ser una transmisión continua, una transmisión de descarga o una combinación de ambas.

[0035] Las técnicas de esta divulgación no están limitadas necesariamente a aplicaciones o configuraciones inalámbricas. Las técnicas se pueden aplicar a la codificación de vídeo como soporte de cualquiera entre varias aplicaciones de multimedios, tales como difusiones de televisión por el aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones continuas de vídeo por Internet, tales como la transmisión continua adaptativa dinámica por HTTP (DASH), el vídeo digital que se codifica en un medio de almacenamiento de datos, la decodificación de vídeo digital almacenado en un medio de almacenamiento de datos, u otras aplicaciones. En algunos ejemplos, el sistema 10 puede estar configurado para prestar soporte a una transmisión de vídeo unidireccional o bidireccional, a fin de prestar soporte a aplicaciones tales como la transmisión de vídeo en tiempo real, la reproducción de vídeo, la difusión de vídeo y/o la videotelefonía.

[0036] En el ejemplo de la figura 1, el dispositivo de origen 12 incluye un origen de vídeo 18, una unidad de estimación de profundidad 19, un codificador de vídeo 20 y una interfaz de salida 22. El dispositivo de destino 14 incluye la interfaz de entrada 28, el decodificador de vídeo 30, la unidad de representación basada en la imagen de profundidad (DIBR) 31 y el dispositivo de visualización 32. En otros ejemplos, un dispositivo de origen y un dispositivo de destino pueden incluir otros componentes o disposiciones. Por ejemplo, el dispositivo de origen 12 puede recibir datos de vídeo desde un origen de vídeo externo 18, tal como una cámara externa. Del mismo modo, el dispositivo de destino 14 puede interactuar con un dispositivo de visualización externo, en lugar de incluir un dispositivo de visualización integrado.

[0037] El sistema ilustrado 10 de la figura 1 es simplemente un ejemplo. Las técnicas de esta divulgación pueden realizarse mediante cualquier dispositivo de codificación y/o decodificación de vídeo digital. Aunque, en general, las técnicas de la presente divulgación son llevadas a cabo por un dispositivo de codificación de vídeo, las técnicas también pueden ser llevadas a cabo por un codificador/decodificador de vídeo, denominado habitualmente "CÓDEC". Además, las técnicas de la presente divulgación también pueden ser llevadas a cabo 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 codificados para su transmisión al dispositivo de destino 14. En algunos ejemplos, los dispositivos 12, 14 pueden funcionar de manera esencialmente simétrica, de modo que cada uno de los dispositivos 12, 14 incluya componentes de codificación y decodificación de vídeo. De ahí que el sistema 10 pueda prestar soporte a la transmisión de vídeo unidireccional o bidireccional entre los dispositivos de vídeo 12, 14, por ejemplo, para la transmisión continua de vídeo, la reproducción de vídeo, la difusión de vídeo o la videotelefonía.

[0038] El origen de vídeo 18 del dispositivo de origen 12 puede incluir un dispositivo de captura de vídeo, tal como una cámara de vídeo, un archivo de vídeo que contiene vídeo capturado previamente y/o una interfaz de alimentación de vídeo para recibir vídeo desde un proveedor de contenidos de vídeo. Como una alternativa adicional, el origen de vídeo 18 puede generar datos basados en gráficos por ordenador como el vídeo de origen, o una combinación de vídeo en directo, vídeo archivado y vídeo generado por ordenador. En algunos casos, si el origen de vídeo 18 es una videocámara, el dispositivo de origen 12 y el dispositivo de destino 14 pueden formar los llamados teléfonos con cámara o videoteléfonos. Sin embargo, como se ha mencionado anteriormente, las técnicas descritas en la presente divulgación pueden ser aplicables a la codificación de vídeo en general, y se pueden aplicar a aplicaciones inalámbricas y/o cableadas. En cada caso, el codificador de vídeo 20 puede codificar el vídeo capturado, precapturado o generado por ordenador. La información de vídeo codificada puede ser emitida, a continuación, por la interfaz de salida 22 hacia un medio legible por ordenador 16.

[0039] El origen de vídeo 18 puede proporcionar múltiples vistas de datos de vídeo al codificador de vídeo 20. Por ejemplo, el origen de vídeo 18 puede corresponder a una formación de cámaras, teniendo cada una posición horizontal única con respecto a una escena particular que se está filmando. De forma alternativa, el origen de vídeo 18 puede generar datos de vídeo desde perspectivas dispares de cámara horizontal, por ejemplo, usando gráficos por ordenador. La unidad de estimación de profundidad 19 puede configurarse para determinar valores para píxeles de profundidad correspondientes a píxeles en una imagen de textura. Por ejemplo, la unidad de estimación de profundidad 19 puede representar una unidad de Navegación y Medición por Sonido (SONAR), una unidad de Detección y Medición de Luz (LIDAR) u otra unidad capaz de determinar directamente los valores de profundidad de forma esencialmente simultánea a la grabación de datos de vídeo de una escena.

[0040] De forma adicional o alternativa, la unidad de estimación de profundidad 19 puede configurarse para calcular valores de profundidad de manera indirecta mediante la comparación de dos o más imágenes que se capturaron esencialmente al mismo tiempo desde diferentes perspectivas de cámara horizontal. Calculando la disparidad horizontal entre valores de píxeles esencialmente similares en las imágenes, la unidad de estimación de profundidad 19 puede aproximarse a la profundidad de varios objetos en la escena. La unidad de estimación de profundidad 19 puede estar integrada funcionalmente con el origen de vídeo 18, en algunos ejemplos. Por ejemplo, cuando el origen de vídeo 18 genera imágenes de gráficos por ordenador, la unidad de estimación de profundidad 19 puede proporcionar mapas de profundidad reales para objetos gráficos, por ejemplo, usando coordenadas z de píxeles y objetos usados para representar imágenes de textura.

[0041] El medio legible por ordenador 16 puede incluir medios transitorios, tales como una difusión inalámbrica o una transmisión de red cableada, o medios de almacenamiento (es decir, medios de almacenamiento no transitorio), tales como un disco duro, una unidad de memoria flash, un disco compacto, un disco de vídeo digital, un disco Blu-ray u otros medios legibles por ordenador. En algunos ejemplos, un servidor de red (no mostrado) puede recibir datos de vídeo codificados desde el dispositivo de origen 12 y proporcionar los datos de vídeo codificados al dispositivo de destino 14, por ejemplo, mediante transmisión por red. De forma similar, un dispositivo informático de una utilidad de producción de un medio, tal como una utilidad de grabación de discos, puede recibir datos de vídeo codificados desde el dispositivo de origen 12 y producir un disco que contiene los datos de vídeo codificados. Por lo tanto, se puede entender que el medio legible por ordenador 16 incluye uno o más medios legibles por ordenador de diversas formas, en diversos ejemplos.

[0042] La interfaz de entrada 28 del dispositivo de destino 14 recibe información desde el medio legible por ordenador 16. La información del medio legible por ordenador 16 puede incluir información sintáctica definida por el codificador de vídeo 20, que también usa el decodificador de vídeo 30, que incluye elementos sintácticos que describen características y/o procesamiento de bloques y otras unidades codificadas, por ejemplo, los GOP. El dispositivo de visualización 32 muestra los datos de vídeo decodificados a un usuario, y puede comprender cualquiera entre varios dispositivos de visualización, tales como un tubo de rayos catódicos (CRT), una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de visualización. En algunos ejemplos, el dispositivo de visualización 32 puede comprender un dispositivo capaz de mostrar dos o más vistas de forma simultánea o esencialmente simultánea, por ejemplo, para producir un efecto visual tridimensional para un espectador.

[0043] La unidad de DIBR 31 del dispositivo de destino 14 puede representar vistas sintetizadas usando información de textura y profundidad de vistas decodificadas, recibidas desde el decodificador de vídeo 30. Por ejemplo, la unidad de DIBR 31 puede determinar la disparidad horizontal para datos de píxeles de imágenes de textura como una función de los valores de píxeles en mapas de profundidad correspondientes. La unidad de DIBR 31 puede generar entonces una imagen sintetizada mediante el desplazamiento de píxeles en una imagen de textura, a la izquierda o a la derecha, en la disparidad horizontal determinada. De esta manera, el dispositivo de visualización 32 puede mostrar una o más vistas, que pueden corresponder a vistas decodificadas y/o vistas sintetizadas, en cualquier combinación. El decodificador de vídeo 30 puede proporcionar valores de precisión para los rangos de profundidad y parámetros de cámara a la unidad de DIBR 31, que puede usar los rangos de profundidad y los parámetros de la cámara para sintetizar adecuadamente las vistas.

[0044] Aunque no se muestra en la figura 1, en algunos aspectos, tanto el codificador de vídeo 20 como el

decodificador de vídeo 30 pueden estar integrados con un codificador y decodificador de audio, y pueden incluir unidades MUX-DEMUX adecuadas, u otro tipo de hardware y software, para ocuparse de la codificación tanto de audio como de vídeo en un flujo de datos común o en flujos de datos separados. Si procede, las unidades MUX-DEMUX se pueden ajustar al protocolo multiplexador H.223 de la ITU o a otros protocolos tales como el protocolo de datagramas de usuario (UDP).

[0045] El codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar de acuerdo a una norma de codificación de vídeo, tal como la norma de codificación de vídeo de alta eficacia (HEVC), actualmente en fase de desarrollo, y pueden ajustarse al modelo de prueba de la HEVC (HM). De forma alternativa, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar de acuerdo a otras normas privadas o industriales, tales como la norma ITU-T H.264, alternativamente denominada MPEG-4, Parte 10, codificación de vídeo avanzada (AVC), o ampliaciones de dichas normas, como la ampliación MVC de ITU-T H.264/AVC.

[0046] Tanto el codificador de vídeo 20 como el decodificador de vídeo 30 pueden implementarse como cualquiera entre varios circuitos codificadores adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), formaciones de compuertas programables in situ (FPGA), lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. Cuando las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio adecuado no transitorio legible por ordenador, y ejecutar las instrucciones en hardware usando uno o más procesadores para realizar las técnicas de esta divulgación. Cada uno entre el codificador de vídeo 20 y el decodificador de vídeo 30 se puede incluir en uno o más codificadores o decodificadores, cualquiera de los cuales se puede integrar como parte de un codificador/decodificador (CÓDEC) combinado en un dispositivo respectivo. Un dispositivo que incluye el codificador de vídeo 20 y/o el decodificador de vídeo 30 puede comprender un circuito integrado, un microprocesador y/o un dispositivo de comunicación inalámbrica, tal como un teléfono celular.

[0047] Inicialmente, se analizarán las técnicas ejemplares de codificación de la HEVC. El WD10 de la HEVC, mencionado anteriormente, supone varias capacidades adicionales de los dispositivos de codificación de vídeo respecto a los dispositivos existentes, de acuerdo, por ejemplo, a la norma ITU-T H.264/AVC. Por ejemplo, mientras que la norma H.264 proporciona nueve modalidades de codificación de intrapredicción, el HM puede proporcionar hasta treinta y tres modalidades de codificación de intrapredicción angulares, más modalidades DC y planares.

[0048] En el borrador WD10 de la norma HEVC y en otras normas de codificación de vídeo, una secuencia de vídeo incluye típicamente una serie de imágenes. Las imágenes también pueden denominarse "tramas". Una imagen puede incluir tres formaciones de muestras, denotadas S_L, S_{Cb} y S_{Cr}, S_L es una formación bidimensional (es decir, un bloque) de muestras de luma. S_{Cb} es una formación bidimensional de muestras de crominancia Cb. S_{Cr} es una formación bidimensional de muestras de crominancia Cr. Las muestras de crominancia también se pueden denominar en el presente documento muestras de "croma". En otros casos, una imagen puede ser monocromática y puede solo incluir una formación de muestras de luma.

[0049] El documento WD10 de la HEVC describe que una trama o imagen de vídeo puede dividirse en una secuencia de unidades arboladas de codificación (CTU) o unidades máximas de codificación (LCU), que incluyen muestras tanto de luma como de croma. Una CTU tiene un fin similar al de un macrobloque de la norma H.264. Un fragmento incluye un cierto número de CTU consecutivas en orden de codificación. Una trama o imagen de vídeo puede dividirse en uno o más fragmentos. Cada CTU puede dividirse en unidades de codificación (CU) de acuerdo a un árbol cuádruple. Por ejemplo, una CTU, como un nodo raíz del árbol cuádruple, puede dividirse en cuatro nodos hijos, y cada nodo hijo puede, a su vez, ser un nodo padre y dividirse en otros cuatro nodos hijos. Un nodo hijo final, no dividido, como un nodo hoja del árbol cuádruple, comprende un nodo de codificación, es decir, un bloque de vídeo codificado. Los datos sintácticos asociados a un flujo de bits codificado pueden definir un número máximo de veces en que puede dividirse una CTU, y también pueden definir un tamaño mínimo de los nodos de codificación.

[0050] Una CU incluye un nodo de codificación y unidades de predicción (PU) y unidades de transformación (TU) asociadas al nodo de codificación. Un tamaño de la CU corresponde a un tamaño del nodo de codificación y debe ser de forma cuadrada. El tamaño de la CU puede variar desde 8x8 píxeles hasta el tamaño del bloque arbolado, con un máximo de 64x64 píxeles o más. Cada CU puede contener una o más PU y una o más TU. Los datos sintácticos asociados a una CU pueden describir, por ejemplo, la división de la CU en una o más PU. Las modalidades de división pueden diferir entre si la CU está codificada en modalidad de salto o directa, codificada en modalidad de intrapredicción o codificada en modalidad de interpredicción. Las PU pueden dividirse para tener forma no cuadrada. Los datos sintácticos asociados a una CU también pueden describir, por ejemplo, la división de la CU en una o más TU de acuerdo con un árbol cuádruple. Una TU puede tener forma cuadrada o no cuadrada.

[0051] El borrador WD10 de la norma HEVC admite transformaciones de acuerdo a las TU, que pueden ser diferentes para diferentes CU. El tamaño de las TU se basa habitualmente en el tamaño de las PU dentro de una CU definida dada para una CTU o LCU dividida, aunque puede que no sea siempre el caso. Las TU son habitualmente del mismo tamaño o de un tamaño más pequeño que las PU. En algunos ejemplos, las muestras residuales correspondientes a una CU pueden subdividirse en unidades más pequeñas utilizando una estructura de árbol cuádruple conocida como «árbol cuádruple residual» (RQT). Los nodos hoja del RQT pueden denominarse TU. Los

valores de diferencias de píxeles asociados a las TU pueden transformarse para generar coeficientes de transformación, que pueden cuantizarse.

[0052] En general, una PU incluye datos relacionados con el proceso de predicción. Por ejemplo, cuando la PU se codifica en intramodalidad, la PU puede incluir datos que describen una modalidad de intrapredicción para la PU. En otro ejemplo, cuando la PU se codifica en intermodalidad, la PU puede incluir datos que definen un vector de movimiento para la PU. Los datos que definen el vector de movimiento para una PU pueden describir, por ejemplo, un componente horizontal del vector de movimiento, un componente vertical del vector de movimiento, una resolución para el vector de movimiento (por ejemplo, precisión de 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.

[0053] En general, se usa una TU para los procesos de transformación y cuantización. Una CU dada que tiene una o más PU también puede incluir una o más TU. Tras la predicción, el codificador de vídeo 20 puede calcular valores residuales correspondientes a la PU. Los valores residuales comprenden valores de diferencias de píxeles que pueden ser transformados en coeficientes de transformación, cuantizados y explorados usando las TU, para generar coeficientes de transformación en serie para la codificación por entropía. Esta divulgación usa habitualmente el término «bloque de vídeo» para referirse a un nodo de codificación de una CU. En algunos casos específicos, esta divulgación también puede usar el término «bloque de vídeo» para referirse a una CTU, también mencionada como una LCU, o una CU, que incluye un nodo de codificación y unas PU y TU.

[0054] Una secuencia de vídeo incluye habitualmente una serie de tramas o imágenes de vídeo. Un grupo de imágenes (GOP) comprende, en general, una serie de una o más de las imágenes de vídeo. Un GOP puede incluir datos sintácticos en una cabecera del GOP, una cabecera de una o más de las imágenes, o en otras ubicaciones, que describen una serie de imágenes incluidas en el GOP. Cada fragmento de una imagen puede incluir datos sintácticos de fragmento que describen una modalidad de codificación para el fragmento respectivo. El codificador de vídeo 20 actúa habitualmente sobre bloques de vídeo dentro de fragmentos de vídeo individuales con el fin de codificar los datos de vídeo. Un bloque de vídeo puede corresponder a un nodo de codificación dentro de una CU. Los bloques de vídeo pueden tener tamaños fijos o variables y pueden diferir en tamaño de acuerdo a una norma de codificación especificada.

[0055] En un ejemplo, el borrador WD10 de la HEVC presta soporte a la predicción en diversos tamaños de PU. Si se supone que el tamaño de una CU particular es 2Nx2N, el borrador WD10 de la norma HEVC presta soporte a la intrapredicción en tamaños de PU de 2Nx2N o NxN, y a la interpredicción en tamaños de PU simétricas de 2Nx2N, 2NxN, Nx2N o NxN. El borrador WD10 de la norma HEVC también presta soporte a la división asimétrica para la interpredicción en tamaños de PU de 2NxnU, 2NxnD, nLx2N y nRx2N. En la división asimétrica, una dirección de una CU no está dividida, mientras que la otra dirección está dividida en un 25 % y un 75 %. La parte de la CU correspondiente a la división del 25 % está indicada por una «n» seguida por una indicación de «arriba», «abajo», «izquierda» o «derecha». Así pues, por ejemplo, «2NxnU» se refiere a una CU de tamaño 2Nx2N que está dividida horizontalmente, con una PU de tamaño 2Nx0,5N encima y una PU de tamaño 2Nx1,5N debajo.

[0056] En la presente divulgación, «NxN» y «N por N» pueden usarse indistintamente para hacer referencia a las dimensiones de píxeles de un bloque de vídeo en términos de dimensiones verticales y horizontales, por ejemplo, 16x16 píxeles o 16 por 16 píxeles. En general, un bloque de 16x16 tendrá 16 píxeles en una dirección vertical (y = 16) y 16 píxeles en una dirección horizontal (x = 16). Asimismo, un bloque de tamaño 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.

[0057] Tras la codificación intrapredictiva o interpredictiva usando las PU de una CU, el codificador de vídeo 20 puede calcular datos residuales para las TU de la CU. Las PU pueden comprender datos de píxeles en el dominio espacial (también denominado dominio del píxel) y las TU pueden comprender coeficientes en el dominio de la transformación tras la aplicación de una transformación, por ejemplo, una transformación discreta de coseno (DCT), una transformación de enteros, una transformación de ondículas o una transformación conceptualmente similar a los datos de vídeo residuales. Los datos residuales pueden corresponder a diferencias de píxeles entre píxeles de la imagen no codificada y valores de predicción correspondientes a las PU. El codificador de vídeo 20 puede formar las TU incluyendo los datos residuales para la CU y, a continuación, transformar las TU para generar coeficientes de transformación para la CU.

[0058] Tras cualquier transformación para generar coeficientes de transformación, el codificador de vídeo 20 puede realizar una cuantización de los coeficientes de transformación. La cuantización se refiere, en general, a un proceso en el que los coeficientes de transformación se cuantizan para reducir posiblemente la cantidad de datos usados para representar los coeficientes, proporcionando compresión adicional. El proceso de cuantización puede reducir la profundidad de bits asociada a algunos, o a la totalidad, de los coeficientes. Por ejemplo, un valor de n bits puede redondearse hacia abajo hasta un valor de m bits durante la cuantización, donde *n* es mayor que *m*.

[0059] En algunos ejemplos, el codificador de vídeo 20 puede utilizar un orden de exploración predefinido para explorar los coeficientes de transformación cuantizados, para producir un vector en serie que se pueda someter a la codificación por entropía. En otros ejemplos, el codificador de vídeo 20 puede realizar una exploración adaptativa. Después de explorar los coeficientes de transformación cuantizados para formar un vector unidimensional, el codificador de vídeo 20 puede realizar la codificación por entropía del vector unidimensional, por ejemplo, de acuerdo a la codificación de longitud variable 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 de división de intervalos de probabilidad (PIPE) u otra metodología de codificación por entropía. El codificador de vídeo 20 también puede realizar la codificación por entropía de los elementos sintácticos asociados a los datos de vídeo codificados, para su uso por el decodificador de vídeo 30 en la decodificación de los datos de vídeo.

[0060] Para realizar la codificación CABAC, el codificador de vídeo 20 puede asignar un contexto dentro de un modelo de contexto a un símbolo a transmitir. El contexto puede relacionarse, por ejemplo, con 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 a transmitir. Las palabras de código en la VLC se pueden construir de modo que los códigos relativamente más cortos correspondan a símbolos más probables, mientras que los códigos más largos correspondan a símbolos menos probables. De esta forma, el uso de la VLC puede lograr un ahorro en bits con respecto, por ejemplo, al uso de palabras de código de igual longitud para cada símbolo a transmitir. La determinación de la probabilidad se puede basar en un contexto asignado al símbolo.

[0061] El codificador de vídeo 20 puede emitir un flujo de bits que incluye una secuencia de bits que forma una representación de imágenes codificadas y datos asociados. El flujo de bits puede comprender una secuencia de unidades de capas de abstracción de red (NAL). Cada una de las unidades de NAL incluye un encabezado de unidad de NAL y encapsula una carga útil de secuencias de octetos sin procesar (RBSP). La cabecera de la unidad de NAL puede incluir un elemento sintáctico que indique un código de tipo de unidad de NAL. El código de tipo de unidad de NAL especificado por la cabecera de unidad de NAL de una unidad de NAL indica el tipo de la unidad de NAL. Una RBSP puede ser una estructura sintáctica que contenga un número entero de octetos que se encapsule dentro de una unidad de NAL. En algunos casos, una RBSP incluye cero bits.

[0062] Diferentes tipos de unidades de NAL pueden encapsular diferentes tipos de RBSP. Por ejemplo, un primer tipo de unidad de NAL puede encapsular una RBSP para un conjunto de parámetros de imagen (PPS), un segundo tipo de unidad de NAL puede encapsular una RBSP para un fragmento codificado, un tercer tipo de unidad de NAL puede encapsular una RBSP para información de mejora suplementaria (SEI), y así sucesivamente. Un PPS es una estructura sintáctica que puede contener elementos sintácticos que se aplican a cero o más imágenes codificadas completas. Las unidades de NAL que encapsulan las RBSP para datos de codificación de vídeo (a diferencia de las RBSP para conjuntos de parámetros y mensajes de SEI) pueden denominarse unidades de NAL de la capa de codificación de vídeo (VCL). Una unidad de NAL que encapsula un fragmento codificado se puede mencionar aquí como una unidad de NAL de fragmento codificado. Una RBSP para un fragmento codificado puede incluir un encabezado de fragmento y datos de fragmento.

[0063] El decodificador de vídeo 30 puede recibir un flujo de bits. Además, el decodificador de vídeo 30 puede analizar sintácticamente el flujo de bits para decodificar elementos sintácticos del flujo de bits. El decodificador de vídeo 30 puede reconstruir las imágenes de los datos de vídeo basándose, al menos en parte, en los elementos sintácticos decodificados a partir del flujo de bits. El proceso para reconstruir los datos de vídeo puede ser, en general, recíproco al proceso realizado por el codificador de vídeo 20. Por ejemplo, el decodificador de vídeo 30 puede usar vectores de movimiento de las PU para determinar bloques predictivos para las PU de una CU actual. El decodificador de vídeo 30 puede usar un vector de movimiento o vectores de movimiento de las PU para generar bloques predictivos para las PU.

[0064] Además, el decodificador de vídeo 30 puede cuantizar inversamente bloques de coeficientes asociados a las TU de la CU actual. El decodificador de vídeo 30 puede realizar transformaciones inversas en los bloques de coeficientes para reconstruir los bloques de transformación asociados a las TU de la CU actual. El decodificador de vídeo 30 puede reconstruir los bloques de codificación de la CU actual añadiendo las muestras predictivas de los bloques predictivos para las PU de la CU actual a las correspondientes muestras residuales de los bloques de transformación de las TU de la CU actual. Mediante la reconstrucción de los bloques de codificación para cada CU de una imagen, el decodificador de vídeo 30 puede reconstruir la imagen. El decodificador de vídeo 30 puede almacenar imágenes decodificadas en un almacén temporal de imágenes decodificadas, para su salida y/o para su uso en la decodificación de otras imágenes.

[0065] En las normas MV-HEVC y 3D-HEVC, un codificador de vídeo puede generar un flujo de bits de múltiples capas que comprenda una serie de unidades de la capa de abstracción de red (NAL). Diferentes unidades de NAL del flujo de bits pueden estar asociadas a diferentes capas del flujo de bits. Una capa se puede definir como un conjunto de unidades de NAL de capa de codificación de vídeo (VCL) y unidades asociadas de NAL no de VCL que tengan el mismo identificador de capa. Una capa puede ser equivalente a una vista en la codificación de vídeo de múltiples

vistas. En la codificación de vídeo de múltiples vistas, una capa puede contener todos los componentes de vista de la misma capa con diferentes instancias temporales. Cada componente de vista puede ser una imagen codificada de la escena de vídeo que pertenece a una vista específica en un instante de tiempo específico. En la norma 3D-HEVC, una capa puede contener todas las imágenes de profundidad codificadas de una vista específica o imágenes de texturas codificadas de una vista específica. De forma similar, en el contexto de la codificación de vídeo ajustable a escala, una capa corresponde habitualmente a imágenes codificadas que tienen características de vídeo diferentes a las de imágenes codificadas en otras capas. Dichas características de vídeo habitualmente incluyen la resolución espacial y el nivel de calidad (razón entre señal y ruido). En la norma HEVC y en sus ampliaciones, la ajustabilidad temporal a escala se puede lograr dentro de una capa definiendo un grupo de imágenes con un nivel temporal particular como una subcapa.

[0066] Para cada capa respectiva del flujo de bits, los datos en una capa inferior pueden decodificarse sin referencia a unos datos en cualquier capa superior. En la codificación de vídeo ajustable a escala, por ejemplo, los datos en una capa base pueden decodificarse sin referencia a datos en una capa de mejora. Las unidades de NAL solo encapsulan datos de una única capa. Una vista puede denominarse "capa base" si el decodificador de vídeo 30 puede decodificar imágenes en la vista sin referencia a datos de cualquier otra capa. La capa base puede ajustarse a la especificación básica de la HEVC. Por tanto, las unidades de NAL que encapsulan datos de la capa restante más alta del flujo de bits pueden eliminarse del flujo de bits sin afectar a la decodificabilidad de los datos en las capas restantes del flujo de bits. En las normas MV-HEVC y 3D-HEVC, las capas superiores pueden incluir componentes de vista adicionales.

[0067] Algunas características generales de la codificación de vídeo serán expuestas a continuación con más detalle. Inicialmente, se expondrá la información de movimiento. Para cada bloque, un conjunto de información de movimiento puede estar disponible. Un conjunto de información de movimiento incluye información de movimiento para las direcciones de predicción hacia adelante y hacia atrás. Aquí, las direcciones de predicción hacia adelante y hacia atrás son dos direcciones de predicción de una modalidad de predicción bidireccional y los términos "hacia adelante" y "hacia atrás" no tienen necesariamente un significado geométrico; en cambio, corresponden a la lista de imágenes de referencia 0 (ListaImgRef0) y a la lista de imágenes de referencia 1 (ListaImgRef1) de una imagen actual. Cuando solo se dispone de una lista de imágenes de referencia para una imagen o fragmento, solo puede estar disponible la ListaImgRef0 y la información de movimiento de cada bloque de un fragmento siempre es hacia delante.

[0068] Para cada dirección de predicción, la información de movimiento incluye un índice de referencia y un vector de movimiento. En algunos casos, por simplicidad, se puede hacer referencia a un vector de movimiento por sí mismo, de manera que se suponga que tiene un índice de referencia asociado. Se utiliza un índice de referencia para identificar una imagen de referencia en la lista actual de imágenes de referencia (ListaImgRef0 o ListaImgRef1). Un vector de movimiento tiene un componente horizontal y otro vertical.

[0069] El recuento de orden de imagen (POC) se usa ampliamente en las normas de codificación de vídeo para identificar el orden de visualización de una imagen. Aunque hay casos en los que dos imágenes dentro de una secuencia de vídeo codificado pueden tener el mismo valor de POC, por lo general no ocurre dentro de una secuencia de vídeo codificado. Cuando están presentes múltiples secuencias de vídeo codificadas en un flujo de bits, las imágenes con un mismo valor de POC pueden estar más cercanas entre sí en términos de orden de decodificación. Los valores de POC de las imágenes se utilizan normalmente para la construcción de la lista de imágenes de referencia, la obtención de un conjunto de imágenes de referencia como en la HEVC y el ajuste a escala del vector de movimiento.

[0070] Algunas técnicas específicas en la norma H.264/AVC serán expuestas a continuación. Inicialmente, se expondrá la estructura de macrobloques en la AVC. En la norma H.264/AVC, cada inter-macrobloque (MB) se puede dividir en cuatro formas diferentes: una partición de MB de tamaño 16x16, dos particiones de MB de tamaño 16x8, dos particiones de MB de tamaño 8x16 o cuatro particiones de MB de tamaño 8x8. Las diferentes particiones de MB en un MB pueden tener diferentes valores del índice de referencia para cada dirección (es decir, ListaImgRef0 o ListaImgRef1). Cuando un MB no está dividido en cuatro particiones de MB de tamaño 8x8, el MB tiene sólo un vector de movimiento para toda la partición de MB en cada dirección. Cuando un MB se divide en cuatro particiones de MB de tamaño 8x8, cada partición de MB de tamaño 8x8 puede dividirse además en subbloques, cada uno de los cuales puede tener un vector de movimiento diferente en cada dirección.

[0071] Existen cuatro formas diferentes de obtener subbloques a partir de una partición de MB de tamaño 8x8: un subbloque de tamaño 8x8, dos subbloques de tamaño 8x4, dos subbloques de tamaño 4x8 o cuatro subbloques de tamaño 4x4. Cada subbloque puede tener un vector de movimiento diferente en cada dirección. Los vectores de movimiento, por lo tanto, están presentes en un nivel igual o superior al nivel de subbloque.

[0072] Se expondrá a continuación la modalidad directa temporal en la AVC. En la AVC, la modalidad directa temporal puede habilitarse en el nivel de MB o bien en el nivel de partición de MB para las modalidades de salto o directa en los fragmentos B. Para cada partición de MB, los vectores de movimiento del bloque cosituados con la partición de MB actual en la ListaImgRef1[0] del bloque actual se utilizan para obtener los vectores de movimiento. Cada vector de movimiento en el bloque cosituado se ajusta a escala en función de las distancias de POC.

[0073] Algunas técnicas específicas en la HEVC se expondrán a continuación. En la HEVC, la unidad de codificación (CU) más grande en un fragmento se denomina bloque arbolado de codificación (CTB). Un CTB incluye un árbol cuádruple, y los nodos del árbol cuádruple son las CU. El tamaño de un CTB puede variar de 16x16 a 64x64 en el perfil principal de la HEVC, aunque, técnicamente, se puede prestar soporte a CTB de tamaños de 8x8. El tamaño de una CU puede variar desde uno tan grande como el mismo tamaño que el CTB hasta uno tan pequeño como 8x8. Cada CU está codificada con una modalidad. Cuando una CU está intercodificada, la CU puede dividirse adicionalmente en dos o cuatro unidades de predicción (PU) o convertirse en una única PU cuando no se aplican particiones adicionales. Cuando la CU está intercodificada, está presente un conjunto de información de movimiento para cada PU de la CU. Además, cada PU de la CU se codifica con una modalidad única de interpredicción para obtener el conjunto asociado de información de movimiento.

[0074] La figura 2 es un diagrama conceptual que ilustra ocho modalidades de partición 33A a 33H ("modalidades de partición 33") para definir las PU de una CU. Una PU es una región en la que se aplica la predicción que se define mediante la partición de la CU. Como se muestra en la figura 2, una PU no se limita a ser de forma cuadrada, para facilitar la partición de la CU para que coincida con las fronteras de los objetos reales en la imagen. Cada CU incluye una, dos o cuatro PU en función de la modalidad de partición. En algunos ejemplos, cuando dos PU están presentes en una CU, las PU pueden ser rectángulos de medio tamaño, como en las modalidades de partición PART_2NxN 33B y PART_Nx2N 33C. En otros ejemplos, cuando dos PU están presentes en una CU, las PU pueden ser dos rectángulos con tamaños de un cuarto o tres cuartos del tamaño de la CU, como en las modalidades de partición PART_2NxnU 33E, PART_2NxnD 33F, PART_nLx2N 33G y PART_nRx2N 33H. Las modalidades de partición PART_2Nx2N 33A y PART_NxN 33D se utilizan para una CU intracodificada. La modalidad de partición PART_NxN 33D solo se permite cuando el tamaño de CU correspondiente es igual al tamaño mínimo de CU.

[0075] La figura 3 es un diagrama conceptual que ilustra una estructura de árbol de transformación 34 dentro de una CU. Cada CU corresponde a un árbol de transformación, que es un árbol cuádruple, cuya hoja es una unidad de transformación (TU). Cada una de las TU en la estructura del árbol de transformación 34 es una región cuadrada, en la que se aplican los procesos de transformación y cuantización, que está definida por la partición de árbol cuádruple de la CU. Como se muestra en la figura 3, una TU es siempre de forma cuadrada. Un TU puede ser tan grande como de 32x32 muestras, hasta 4x4 muestras. Una profundidad máxima de árbol cuadrático, es decir, hasta cuánto se puede dividir la estructura de árbol de transformación 34 para formar nodos de hoja, es ajustable y se especifica en la sintaxis del encabezado de fragmento. Para una inter-CU, una TU puede ser más grande que una PU, es decir, la TU puede contener fronteras de PU. Sin embargo, para una intra-CU, una TU no puede cruzar las fronteras de PU.

[0076] Un elemento sintáctico rqt_raíz_cbf igual a 1 especifica que una estructura sintáctica árbol_transformación está presente para una CU actual. El elemento sintáctico rqt_raíz_cbf igual a 0 especifica que la estructura sintáctica de árbol de transformación no está presente para la CU actual. Cuando el elemento sintáctico rqt_raíz_cbf no está presente, se deduce que su valor es igual a 1. Cuando el elemento sintáctico rqt_raíz_cbf es igual a 0, el árbol de transformación solo contiene un nodo, lo que significa que no se divide adicionalmente y que un indicador de transformación dividida es igual a 0. Para un nodo dentro del árbol de transformación, si tiene un indicador de transformación dividida igual a 1, se divide adicionalmente en cuatro nodos, y una hoja del árbol de transformación tiene un indicador_transformación_dividida igual a 0.

[0077] Por simplicidad, si una TU o un árbol de transformación corresponde a un bloque que no tiene una transformación, todavía se considera que es una TU o una unidad de transformación porque la jerarquía de la propia transformación todavía existe, incluso aunque la transformación no se aplique. Habitualmente, un bloque salteado de transformación corresponde a una unidad de transformación.

[0078] Un indicador de bloque codificado (cbf) de una TU igual a 1 especifica que la TU contiene uno o más niveles de coeficientes de transformación distintos a cero, es decir, no iguales a 0. Un CBF de una TU igual a 0 especifica que todos los niveles de coeficientes de transformación de la TU son iguales a 0. Se fija un CBF para cada componente de la TU, es decir, se fija un CBF para cada uno de los componentes de luminancia (Y) y crominancia (Cb y Cr).

[0079] En la norma HEVC, hay dos modalidades de interpredicción, denominadas modalidades de fusión y de predicción avanzada de vector de movimiento (AMVP), para predecir el movimiento para una unidad de predicción (PU) actual. Una modalidad de omisión se considera un caso especial de la modalidad de fusión. Tanto en la modalidad de AMVP como en la modalidad de fusión, se mantiene una lista de candidatos de vectores de movimiento (MV) que incluye múltiples predictores de vectores de movimiento. Los vectores de movimiento, así como los índices de referencia asociados en la modalidad de fusión, se generan para la PU actual seleccionando uno de los candidatos de la lista de candidatos de MV.

[0080] En la modalidad de fusión, la lista de candidatos de MV incluye hasta cinco candidatos. Un candidato de fusión puede contener un conjunto de información de movimiento, por ejemplo, vectores de movimiento correspondientes a ambas listas de imágenes de referencia (lista 0 y lista 1) y los índices de referencia asociados. Si se identifica un candidato de fusión mediante un índice de fusión para la PU actual, los vectores de movimiento y las imágenes de referencia asociadas se utilizan para la predicción de la PU actual.

[0081] En la modalidad de AMVP, la lista de candidatos de MV incluye solo dos candidatos. Un candidato a AMVP contiene vectores de movimiento correspondientes a ambas listas de imágenes de referencia (lista 0 y lista 1). Si un candidato de AMVP se identifica con un índice de AMVP para la PU actual, un índice de referencia debe ser señalado explícitamente para cada posible dirección de predicción de la lista 0 o la lista 1, junto con el índice de AMVP, ya que el candidato de AMVP incluye solo un vector de movimiento. En la modalidad de AMVP, los vectores de movimiento predichos se pueden refinar aún más mediante la señalización de una diferencia de vectores de movimiento y realizando un ajuste a escala del vector de movimiento.

[0082] Como se ha descrito anteriormente, un candidato de fusión corresponde a un conjunto completo de información de movimiento, mientras que un candidato de AMVP contiene solo un vector de movimiento para una dirección de predicción específica y un índice de referencia explícitamente señalizado. Los candidatos para ambas modalidades se obtienen de manera similar a partir de los mismos bloques vecinos espaciales y temporales de la PU actual.

[0083] Las figuras 4A y 4B son diagramas conceptuales que ilustran candidatos adyacentes espaciales para la modalidad de fusión y la modalidad de AMVP, respectivamente. Los candidatos de MV espaciales se obtienen de los bloques vecinos que se muestran en las figuras 4A y 4B, para una PU actual (PU0), aunque los procedimientos para generar los candidatos a partir de los bloques difieren para las modalidades de fusión y de AMVP.

[0084] En la modalidad de fusión, se pueden obtener hasta cuatro candidatos de MV espaciales de los bloques vecinos de la PU0 90 en el orden numerado que se muestra en la figura 4A. En el ejemplo ilustrado de la figura 4A, el orden es el siguiente: bloque izquierdo (0), bloque superior (1), bloque superior derecho (2), bloque inferior izquierdo (3) y bloque superior izquierdo (4).

[0085] En la modalidad de AMVP, se pueden obtener hasta dos candidatos de MV espaciales de los bloques vecinos de la PU0 92 en el orden numerado que se muestra en la figura 4B. Los bloques vecinos de la PU0 92 se dividen en dos grupos: un grupo izquierdo que incluye el bloque inferior izquierdo (0) y el bloque izquierdo (1), y un grupo superior que incluye el bloque derecho superior (2), el bloque superior (3) y el bloque izquierdo superior (4), como se muestra en la figura 4B. Para cada grupo, un vector de movimiento para un bloque vecino que se refiere a la misma imagen de referencia que el índice de referencia señalizado explícitamente para la PU0 92 tiene la máxima prioridad para ser elegido para formar un candidato de MV final del grupo. En algunos ejemplos, es posible que ninguno de los bloques vecinos tenga un vector de movimiento que apunte a la imagen de referencia explícitamente señalizada. Si no se puede encontrar un candidato de MV de ese tipo, el primer vector de movimiento disponible se ajustará a escala para formar el candidato de MV final del grupo. En este caso, cualquier diferencia de distancia temporal será compensada.

[0086] La figura 5A es un diagrama conceptual que ilustra los candidatos de predicción de vectores de movimiento temporal (TMVP) para una PU actual (PU0) 94. Un candidato de TMVP, si está habilitado y disponible, se agrega a la lista de candidatos de MV después de los candidatos de MV espacial. El proceso de obtención de vectores de movimiento para un candidato de TMVP es el mismo tanto para la modalidad de fusión como para las modalidades de AMVP. Sin embargo, en la modalidad de fusión, el índice de referencia de destino para el candidato de TMVP siempre se fija en 0.

[0087] La ubicación del bloque primario para la obtención de candidatos de TMVP es el bloque inferior derecho fuera de una PU (PU1) cosituada 96 de la PU0 94, como se muestra en la figura 5A como bloque T 97. El bloque inferior derecho T 97 se selecciona para compensar el sesgo de los bloques superiores e izquierdos utilizados para generar los candidatos espaciales adyacentes. Sin embargo, si el bloque T inferior derecho está ubicado fuera de la LCU 98 actual de la PU0 94 (es decir, en la posición 99) o la información de movimiento para el bloque inferior derecho T 97 no está disponible, el bloque utilizado para la obtención de candidatos de TMVP es sustituido por un bloque central T95 de la PU0 94.

[0088] La figura 5B es un diagrama conceptual que ilustra el ajuste a escala de los MV para la modalidad de TMVP para una PU actual 100 en la imagen actual 102. Un vector de movimiento para un candidato de TMVP se obtiene de una PU 104 de ubicación conjunta de la imagen 106 de ubicación conjunta, indicado en el nivel de fragmento. De manera similar a la modalidad directa temporal en la norma AVC, descrita anteriormente, se puede generar un vector de movimiento del candidato de TMVP para la PU 100 actual usando el ajuste a escala, que se realiza para compensar las diferencias entre la distancia temporal co-localizada (es decir, la distancia entre la imagen co-localizada 106 y la imagen de referencia co-localizada 107) y la diferencia temporal actual (es decir, entre la imagen actual 102 y la imagen de referencia actual 103), como se muestra en la figura 5B.

[0089] A continuación se expondrán otros aspectos de la predicción de vectores de movimiento en la HEVC. En el ajuste a escala de vectores de movimiento, se supone que el valor de los vectores de movimiento es proporcional a la distancia de las imágenes en el tiempo de presentación. Un vector de movimiento asocia dos imágenes: la imagen de referencia y la imagen que contiene el vector de movimiento (es decir, la imagen contenedora). Cuando se utiliza un vector de movimiento para predecir otro vector de movimiento, la distancia de la imagen contenedora y la imagen de referencia se calcula en función de los valores del recuento de orden de imágenes (POC) de las imágenes. Para que un vector de movimiento sea predicho, tanto su imagen contenedora como la imagen de referencia asociada pueden

ser diferentes. En este caso, se calcula una nueva distancia (basada en el POC). El vector de movimiento se ajusta a escala en función de estas dos distancias de POC. Para un candidato vecino espacial, las imágenes contenedoras para los dos vectores de movimiento son las mismas, mientras que las imágenes de referencia son diferentes. En la HEVC, el ajuste a escala del vector de movimiento se aplica tanto a la TMVP como a la AMVP para candidatos vecinos espaciales y temporales.

[0090] En la generación de candidatos de vectores de movimiento artificial, si una lista de candidatos de vectores de movimiento no está completa (es decir, se dispone de menos del número prescrito de candidatos), se puede generar e insertar candidatos de vectores de movimiento artificiales al final de la lista, hasta que la lista incluya el número prescrito de candidatos. En la modalidad de fusión, hay dos tipos de candidatos de VM artificiales: un candidato combinado obtenido solo para fragmentos B y cero candidatos utilizados solo para la AMVP si el primer tipo no proporciona suficientes candidatos artificiales. Para cada par de candidatos que ya están en la lista de candidatos y tienen la información de movimiento necesaria, los candidatos de vectores de movimiento combinados bidireccionales se obtienen basándose en una combinación de un vector de movimiento de un primer candidato que se refiere a una imagen en la lista 0 y un vector de movimiento de un segundo candidato que se refiere a una imagen en la lista 1.

[0091] En el proceso de poda para la inserción de candidatos, puede ocurrir que los candidatos de diferentes bloques sean los mismos, lo que disminuye la eficacia de una lista de candidatos de fusión/AMVP. Se puede aplicar un proceso de poda para abordar este problema. El proceso de poda compara a un candidato con los otros en la lista de candidatos actuales para evitar la inserción de un candidato idéntico en ciertas circunstancias. Para reducir la complejidad, solo se puede aplicar un número limitado de procesos de poda, en lugar de comparar cada candidato potencial con todos los otros candidatos existentes.

[0092] Ahora se describirá un proceso de filtro de desbloqueo en la HEVC. El proceso de filtro de desbloqueo se realiza para cada CU de un bloque de vídeo en el mismo orden que el proceso de decodificación. Primero, se filtran los bordes verticales (es decir, el filtrado horizontal) y luego se filtran los bordes horizontales (es decir, el filtrado vertical). El filtrado de desbloqueo se aplica a las fronteras de bloque dentro de la CU que se determinan para ser filtrados, tanto para componentes de luma como de croma. En algunos ejemplos, las fronteras de bloque de tamaño 8x8 se filtran mientras que las fronteras de bloque de tamaño 4x4 no se procesan, para reducir la complejidad. Dos tipos de fronteras están implicados en el proceso de filtro de desbloqueo: Fronteras de TU y fronteras de PU. Las fronteras de CU también se consideran, ya que las fronteras de CU son necesariamente también fronteras de TU y PU.

[0093] La figura 6 es un diagrama de flujo que ilustra un flujo de procesamiento ejemplar 108 de un proceso de filtro de desbloqueo. Una frontera puede tener tres valores de estado de filtrado: sin filtrado, filtrado débil y filtrado fuerte. Cada decisión de filtrado se basa en la resistencia de frontera, Bs, y en valores de umbral, β y t_C.

[0094] La figura 7 es un diagrama de flujo que ilustra una operación ejemplar de un cálculo de resistencia de frontera (Bs) 110 para un proceso de filtro de desbloqueo. La resistencia de frontera (Bs) refleja cuán potente puede necesitar ser un proceso de filtrado para la frontera. Por ejemplo, un valor de Bs de 2 indica un filtrado potente, un valor de Bs de 1 indica un filtrado débil y un valor de Bs de 0 indica que no hay filtrado de desbloqueo.

[0095] Sean P y Q definidos como bloques que están implicados en el filtrado, donde P representa el bloque ubicado a la izquierda (en el caso del borde vertical) o encima (en el caso del borde horizontal) de la frontera a desbloquear, y Q representa el bloque situado a la derecha (en el caso del borde vertical) o encima (en el caso del borde horizontal) de la frontera a desbloquear. La operación ilustrada en la figura 7 indica cómo se calcula el valor de Bs en función de la modalidad de intracodificación, la existencia de coeficientes de transformación distintos a cero, las imágenes de referencia, el número de vectores de movimiento y las diferencias de vectores de movimiento.

[0096] La figura 8 es un diagrama conceptual que ilustra información reutilizada o referida para un cálculo de Bs entre píxeles en una cuadrícula 112 en una frontera de unidad de árbol de codificación (CTU) 113 para un proceso de filtro de desbloqueo. La Bs se calcula sobre una base de bloques de tamaño 4x4, pero se vuelve a correlacionar con una cuadrícula de tamaño 8x8. El máximo de los dos valores de Bs que corresponden a 8 píxeles, que consisten en una línea en la cuadrícula de tamaño 4x4, se selecciona como la Bs para fronteras en la cuadrícula de tamaño 8x8. En el frontera de CTU 113, la información de cada segundo bloque (en una cuadrícula de tamaño 4x4) a la izquierda o encima se reutiliza como se ilustra en la figura 8, con el fin de reducir el requisito de memoria de almacén temporal de líneas.

[0097] Los valores de umbral β' y t_C' están implicados en la decisión de activación / desactivación del filtro, la selección de filtro potente y débil y el proceso de filtrado débil. Los valores de umbral se obtienen del valor del parámetro de cuantización de luma, Q, como se muestra en la Tabla 1 a continuación.

Tabla 1 - Obtención de variables de umbral a partir de la entrada Q

Q	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
β'	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	7	8
t_C'	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Q	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
β'	9	10	11	12	13	14	15	16	17	18	20	22	24	26	28	30	32	34	36
t_C'	1	1	1	1	1	1	1	1	2	2	2	2	3	3	3	3	4	4	4
Q	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53			
β'	38	40	42	44	46	48	50	52	54	56	58	60	62	64	-	-			
t_C'	5	5	6	6	7	8	9	10	11	13	14	16	18	20	22	24			

[0098] La variable β se obtiene de β' de la siguiente manera:

$$\beta = \beta' * (1 << (ProfundidadBits_Y - 8))$$

La variable t_C se obtiene de t_C' de la siguiente manera:

$$t_C = t_C' * (1 << (ProfundidadBits_Y - 8))$$

[0099] La figura 9 es un diagrama conceptual que ilustra los píxeles en el bloque P 116 y el bloque Q 117 implicados en una decisión de filtro de activación/desactivación y una selección de filtro potente/débil para un proceso de filtro de desbloqueo en la frontera 114. La decisión de activación/desactivación del filtro se realiza utilizando cuatro líneas de píxeles que cruzan la frontera 114, que se agrupan, cada una, como una unidad (que se muestra en la Figura 9 utilizando cuadros trazados en negro) para reducir la complejidad de cálculo. Los seis píxeles incluidos en cada uno de los dos cuadros en un primer conjunto de cuatro líneas 118 se utilizan para determinar si el filtro está activado o desactivado para el primer conjunto de cuatro líneas 118. Los seis píxeles incluidos en cada uno de los dos cuadros en el segundo conjunto de cuatro líneas 119 se utilizan para determinar si el filtro está activado o desactivado para el segundo conjunto de cuatro líneas 119.

[0100] Las siguientes variables se definen para la decisión del filtro de activación / desactivación:

$$dp0 = | p_{2,0} - 2*p_{1,0} + p_{0,0} |$$

$$dp3 = | p_{2,3} - 2*p_{1,3} + p_{0,3} |$$

$$dq0 = | q_{2,0} - 2*q_{1,0} + q_{0,0} |$$

$$dq3 = | q_{2,3} - 2*q_{1,3} + q_{0,3} |$$

Si dp0 + dq0 + dp3 + dq3 < β, se activa el filtrado para el primer conjunto de cuatro líneas 118, y se aplica el proceso de selección de filtro potente/débil. Si esta condición no se satisface, no se realiza ningún filtrado para el primer conjunto de cuatro líneas 118.

[0101] Además, si se satisface la condición, las variables dE, dEp1 y dEp2 se fijan de la siguiente manera:

dE se fija igual a 1

Si dp0 + dp3 < (β + (β >> 1)) >> 3, la variable dEp1 se fija igual a 1

Si dq0 + dq3 < (β + (β >> 1)) >> 3, la variable dEq1 se fija igual a 1

Una decisión de activación/desactivación del filtro se toma de una manera similar a la descrita anteriormente para el segundo conjunto de cuatro líneas 119.

[0102] Si el filtrado está activado, se toma una decisión entre el filtrado potente y el débil. Los píxeles implicados son los mismos que los utilizados para la decisión de activación/desactivación del filtro, como se ilustra en la figura 9. Si se cumplen los siguientes dos conjuntos de condiciones, se utiliza un filtro potente para filtrar el primer conjunto de cuatro líneas 118. De lo contrario, se utiliza un filtro débil.

(1) $2*(dp0+dq0) < (\beta >> 2), | p3_0 - p0_0 | + | q0_0 - q3_0 | < (\beta >> 3)$ y $| p0_0 - q0_0 | < (5* t_C + 1) >> 1$

(2) $2*(dp3+dq3) < (\beta >> 2), | p3_3 - p0_3 | + | q0_3 - q3_3 | < (\beta >> 3)$ y $| p0_3 - q0_3 | < (5* t_C + 1) >> 1$

La decisión de seleccionar un filtrado potente o débil para el segundo conjunto de cuatro líneas 119 se toma de manera similar.

[0103] Para un filtrado potente, los valores de píxeles filtrados se obtienen mediante las siguientes ecuaciones. Téngase en cuenta que tres píxeles se modifican utilizando cuatro píxeles como entrada para cada uno entre el bloque P 116 y bloque Q 117.

$$p_0' = (p_2 + 2*p_1 + 2*p_0 + 2*q_0 + q_1 + 4) >> 3$$

$$q_0' = (p_1 + 2*p_0 + 2*q_0 + 2*q_1 + q_2 + 4) >> 3$$

$$p_1' = (p_2 + p_1 + p_0 + q_0 + 2) >> 2$$

$$q_1' = (p_0 + q_0 + q_1 + q_2 + 2) >> 2$$

$$p_2' = (2*p_3 + 3*p_2 + p_1 + p_0 + q_0 + 4) >> 3$$

$$q_2' = (p_0 + q_0 + q_1 + 3*q_2 + 2*q_3 + 4) >> 3$$

[0104] Para un filtrado débil, los valores de los píxeles filtrados se obtienen mediante las siguientes ecuaciones. Téngase en cuenta que dos píxeles se modifican utilizando tres píxeles como entrada para cada uno entre el bloque P 116 y bloque Q 117.
Delta (Δ) se define de la siguiente manera.

$$\Delta = (9 * (q_0 - p_0) - 3 * (q_1 - p_1) + 8) >> 4$$

Cuando abs (Δ) es menor que t_C *10,

$$\Delta = \text{Clip3}(-t_C, t_C, \Delta)$$

$$p_0' = \text{Clip1}_Y(p_0 + \Delta)$$

$$q_0' = \text{Clip1}_Y(q_0 - \Delta)$$

Si dEp1 es igual a 1,

$$\Delta p = \text{Clip3}(-(t_C >> 1), t_C >> 1, (((p_2 + p_0 + 1) >> 1) - p_1 + \Delta) >> 1)$$

$$p_1' = \text{Clip1}_Y(p_1 + \Delta p)$$

Si dEq1 es igual a 1,

$$\Delta q = \text{Clip3}(-(t_C >> 1), t_C >> 1, (((q_2 + q_0 + 1) >> 1) - q_1 - \Delta) >> 1)$$

$$q_1' = \text{Clip1}_Y(q_1 + \Delta q)$$

[0105] Las decisiones de desbloqueo descritas anteriormente son para filtrar componentes de luma. La resistencia de frontera Bs para el filtrado de croma se hereda de luma. Si Bs > 1, se realiza el filtrado de croma. No se realiza ningún proceso de selección de filtro para el croma, ya que solo se puede aplicar un filtro. Los valores de muestra filtrados p_0' y q_0' se obtienen de la siguiente manera.

$$\Delta = \text{Clip3}(-t_C, t_C, ((((q_0 - p_0) << 2) + p_1 - q_1 + 4) >> 3))$$

$$p_0' = \text{Clip1}_C(p_0 + \Delta)$$

$$q_0' = \text{Clip1}_C(q_0 - \Delta)$$

[0106] Algunas técnicas específicas en la norma 3D-HEVC serán expuestas a continuación. La extensión de vídeo tridimensional de la norma HEVC (3D-HEVC) está siendo desarrollada por JCT-3V. La extensión de la norma 3D-HEVC presta soporte a la codificación de vídeo de múltiples vistas más el formato de profundidad.

[0107] La figura 10 es un diagrama conceptual que ilustra una estructura de predicción de múltiples vistas 120 para casos de 3 vistas, donde V3 indica la vista base y una imagen en una vista no base (es decir, V1 o V5) puede predecirse a partir de imágenes en la vista base de la misma instancia temporal. La predicción de muestras entre vistas, que utiliza la estructura de predicción de múltiples vistas 120, ilustrada en la figura 10, tiene soporte de la extensión de múltiples vistas de la HEVC (MV-HEVC). Tanto la MV-HEVC como la 3D-HEVC son compatibles con la HEVC de manera que la vista de base o de textura (es decir, V3) se pueda decodificar mediante un decodificador de la HEVC.

[0108] En la MV-HEVC, una imagen actual en una vista no base (es decir, V1 o V5) puede ser predicha, tanto por imágenes en la misma vista como por imágenes en una vista de referencia de la misma instancia temporal, incluyendo todas estas imágenes en listas de imágenes de referencia para la imagen actual. Una lista de imágenes de referencia de la imagen actual, por lo tanto, contiene tanto imágenes de referencia temporal como imágenes de referencia entre vistas. Un vector de movimiento asociado a un índice de referencia correspondiente a una imagen de referencia temporal se indica como un vector de movimiento temporal. Un vector de movimiento asociado a un índice de referencia correspondiente a una imagen de referencia entre vistas se indica como un vector de movimiento de disparidad.

[0109] La extensión 3D-HEVC presta soporte a todas las características de la norma MV-HEVC, por lo que la 3D-HEVC también presta soporte a la predicción de muestras entre vistas. Además, la extensión 3D-HEVC presta soporte a herramientas más avanzadas de codificación de solo textura y a herramientas de codificación relacionadas con, o dependientes de, la profundidad, que no tienen soporte por parte de la extensión MV-HEVC. Las herramientas de codificación de solo textura pueden requerir la identificación de los bloques correspondientes entre vistas que pertenecen al mismo objeto. La obtención de vectores de disparidad, por lo tanto, es una tecnología básica en la 3D-HEVC.

[0110] Un aspecto clave de la tecnología de codificación de múltiples vistas es identificar los bloques correspondientes de diferentes vistas mediante una obtención precisa y eficaz de los vectores de disparidad. En la norma 3D-HEVC, la obtención del vector de disparidad basado en bloques vecinos (NBDV) está diseñada de manera similar a las modalidades de AMVP y de fusión en la HEVC. Los vectores de disparidad, sin embargo, se obtienen estrictamente de bloques vecinos, por lo que no se necesitan bits adicionales para un mayor refinamiento o ajuste a escala de los vectores de disparidad. Cuando se habilita la predicción de muestras entre vistas, los vectores de movimiento correspondientes a las imágenes de referencia entre vistas, a saber, los vectores de movimiento de disparidad, ya están presentes junto con los vectores de movimiento normales, a saber, los vectores de movimiento temporales, en el campo de movimiento para un bloque actual. La idea básica de la obtención de los NBDV es hacer uso de los vectores de movimiento de disparidad en el campo de movimiento, verificando solo los bloques vecinos espaciales y temporales.

[0111] La figura 11 es un diagrama conceptual que ilustra los bloques vecinos espaciales a los que se accede para la obtención de los NBDV para un bloque actual 122, donde los bloques vecinos espaciales para la obtención de los NBDV son los mismos a los que se accede para la modalidad de fusión y la modalidad de AMVP. Los bloques adyacentes espaciales, A0, A1, B0, B1 y B2, a los que se accede para los NBDV, son los mismos a los que se accede para las modalidades de AMVP y de fusión, como se muestra en las figuras 4A y 4B y la figura 11.

[0112] Sin embargo, es sumamente posible que no haya ningún bloque espacial adyacente al bloque actual 122 que tenga algunos vectores de movimiento de disparidad asociados. En este caso, también se verifican los bloques vecinos temporales. Los bloques vecinos temporales del bloque actual 122 son los bloques en la imagen de ubicación conjunta, como se usa en la modalidad de TMVP y, además, otra imagen que pueda tener una mejor oportunidad de usar la compensación de movimiento de disparidad, por ejemplo, una imagen de acceso aleatorio o una imagen que tenga un Identificador temporal mínimo. Para cada imagen temporal, de manera similar a la modalidad de TMVP ilustrada en la figura 5A, se comprueban los bloques centrales y los de abajo a la derecha. Todos los bloques vecinos espaciales y temporales del bloque actual 122 se verifican en orden. Una vez que se identifica un vector de movimiento de disparidad, se obtiene el vector de disparidad del bloque actual 122 para que sea el mismo que el vector de movimiento de disparidad identificado, y el proceso de obtención de los NBDV termina. El vector de disparidad del bloque actual 122 se usa para identificar el bloque de referencia entre vistas en la imagen de la vista de referencia, por ejemplo, como en la predicción de movimiento entre vistas y la predicción residual entre vistas.

[0113] En la norma 3D-HEVC, puede ser posible almacenar algunos vectores de disparidad obtenidos como resultados de la obtención de los NBDV, para su uso posterior por los bloques vecinos. Tales vectores de disparidad obtenidos se denominan vectores de disparidad implícita (IDV). Por ejemplo, si la obtención de los NBDV no identifica ningún vector de movimiento de disparidad de los bloques vecinos, los bloques vecinos espaciales se pueden verificar nuevamente para identificar IDV disponibles cualesquiera, de los bloques vecinos espaciales que se usarán para obtener el vector de disparidad final para el bloque actual 122. Recientemente, la 3D-HEVC también incluyó

simplificaciones de los procedimientos de NBDV, verificando menos bloques vecinos espaciales y temporales.

[0114] La predicción de síntesis de vistas (VSP) en la 3D-HEVC se realiza mediante dos módulos técnicos principales: NBDV orientado en profundidad (Do-NBDV) y Predicción de síntesis de retrovisión (BVSP).

[0115] En la obtención de Do-NBDV, el vector de disparidad, generado a partir del esquema de los NBDV, podría refinarse aún más usando la información en el mapa de profundidad codificado. De esta manera, la precisión del vector de disparidad puede mejorarse aprovechando la información codificada en el mapa de profundidad de la vista base. Las etapas de refinamiento se describen de la siguiente manera.

1. Localizar un bloque de profundidad correspondiente basándose en el vector de disparidad obtenido por el esquema de NBDV para una PU actual en una imagen de vista de profundidad de referencia, previamente codificada (tal como en la vista base) de la misma unidad de acceso; el tamaño del bloque de profundidad correspondiente es el mismo que el de la PU actual.

2. Calcular un vector de disparidad para la PU actual a partir del bloque de profundidad correspondiente en función del valor máximo de los cuatro valores de profundidad de píxeles de esquina. El valor máximo se fija igual al componente horizontal del vector de disparidad calculado, mientras que el componente vertical del vector de disparidad se fija en 0.

[0116] El vector de disparidad procedente del esquema de NBDV luego es reemplazado por este vector de disparidad recién obtenido a partir del esquema de Do-NBDV, y el vector de disparidad refinado (de Do-NBDV) se usa para la predicción de movimiento entre vistas para la PU actual. Sin embargo, el vector de disparidad no refinado (a partir de los NBDV) se puede usar para la predicción residual entre vistas para la PU actual. Además, el vector de disparidad refinado puede almacenarse como el vector de movimiento de la PU actual si se codifica en la modalidad de BVSP.

[0117] La modalidad de BVSP se realiza como un candidato de modalidad de fusión especial que tiene su vector de movimiento de inicio configurado para ser un vector de movimiento de disparidad. Dicho candidato se llama un candidato de BVSP. Cuando un bloque vecino espacial de una PU actual se codifica en una modalidad de BVSP, el vector de disparidad asociado y el índice de vista de referencia del bloque vecino se utilizan para definir un candidato de BVSP adicional para la PU actual. El candidato de modalidad de fusión espacial obtenido de un bloque vecino espacial de este tipo se etiqueta con un indicador de BVSP igual a 1.

[0118] La figura 12 es un diagrama conceptual que ilustra una modalidad de BVSP 124 que asigna diferentes vectores de movimiento para diferentes sub-PU de una PU actual 126 en una imagen actual 125. En esta divulgación, la modalidad de BVSP actual en la norma 3D-HEVC puede denominarse modalidad de BVSP de sub-PU. En la modalidad de BVSP, la primera información de movimiento se determina para la PU actual 126 a partir de los bloques vecinos espaciales o temporales de la PU actual 126. Como se ha descrito anteriormente, la primera información de movimiento se puede seleccionar como un candidato de BVSP en una lista de candidatos en modalidad de fusión. La primera información de movimiento incluye un vector de movimiento de disparidad y un índice de vista de referencia asociado que identifica una imagen de referencia entre vistas 128. Esta primera etapa de la modalidad de BVSP puede ser esencialmente similar a la obtención de los NBDV descrita anteriormente.

[0119] La PU actual 126 se divide luego adicionalmente en subregiones o sub-PU. Por ejemplo, una PU con su tamaño indicado por NxM se puede dividir adicionalmente en varias subregiones o sub-PU con tamaños iguales a KxL (donde K y L pueden ser 8 o 4, pero no ambos 4). Para cada subregión o sub-PU, se determina la segunda información de movimiento, que incluye un vector distinto de movimiento de disparidad, obtenido accediendo a un bloque correspondiente entre los bloques de profundidad 130 en una imagen de vista de profundidad de referencia 129, asociada a la imagen de referencia entre vistas 128, identificada por el Índice de vista de referencia asociado al primer vector de disparidad. Más específicamente, para cada sub-PU de la PU actual 126, se selecciona un valor máximo de los cuatro píxeles de esquina del correspondiente bloque entre los bloques de profundidad 130 y se convierte en un componente horizontal del vector de movimiento de disparidad para la sub-PU; el vector de movimiento de disparidad tiene un componente vertical igual a cero. Esta segunda etapa de la modalidad de BVSP puede ser esencialmente similar a la obtención de Do-NBDV, descrita anteriormente.

[0120] Después de que cada subregión o sub-PU de la PU actual 126 tenga predicha su información de movimiento, el motor de compensación de movimiento de la HEVC se puede usar para predecir cada subregión o sub-PU de la PU actual 126 a partir de la imagen de referencia entre vistas 128, basándose en la segunda información de movimiento para cada sub-PU. En la modalidad de BVSP, después de realizar la compensación de movimiento para predecir cada sub-PU de la PU actual 126, solo el primer vector de movimiento de disparidad incluido en la primera información de movimiento, seleccionada para la PU actual 126, se almacena para la PU actual 126, y los distintos vectores de disparidad incluidos en la segunda información de movimiento para las sub-PU se descartan.

[0121] Con la modalidad de BVSP, cada PU puede tener más de un conjunto de información de movimiento, lo que significa que las sub-PU dentro de una PU pueden tener diferentes vectores de movimiento. Esto es diferente de la

HEVC, donde cada PU solo tiene un conjunto de información de movimiento. Habitualmente, en la modalidad de BVSP, aunque las sub-PU dentro de una PU pueden tener diferentes vectores de movimiento, los valores de índices de referencia para las sub-PU son los mismos. El aumento de la complejidad de la modalidad de BVSP se basa principalmente en un acrecentado consumo de energía, y la complejidad del caso más desfavorable es similar a la compensación de movimiento de la HEVC, siempre que el tamaño, así como el estado de predicción doble, de las sub-PU nunca sea menor que el tamaño del bloque de compensación de movimiento habilitado en la HEVC.

[0122] En la norma 3D-HEVC, una vista de profundidad se codifica después de la vista de textura asociada. Cuando se codifica una PU en la vista de profundidad, la información de movimiento de la vista de textura dentro de la región cosituada de la PU puede crear un candidato de fusión para la PU actual. En este caso, se puede hacer referencia al candidato de fusión como el candidato de herencia de parámetros de movimiento (MPI), que contiene un conjunto completo de información de movimiento.

[0123] En el documento JCT3V-E0184, Jicheng An et al., relacionado con la norma 3D-CE3.h: Sub-PU level inter-view motion prediction [Predicción de movimiento entre vistas en el nivel de sub-PU], Equipo de Colaboración Conjunta sobre extensiones de codificación de vídeo tridimensional de la ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 5ª Conferencia: Viena, AT, 27 de julio a 2 de agosto de 2013, se describe el uso de un procedimiento de predicción de movimiento entre vistas en el nivel de sub-PU para el candidato de fusión entre vistas, es decir, el candidato obtenido a partir de un bloque de referencia en una imagen de referencia entre vistas.

[0124] La figura 13 es un diagrama conceptual que ilustra la predicción de movimiento 131 entre vistas en el nivel de sub-PU para una PU actual 134 en una vista actual (V1) 132. Cuando se habilita la modalidad de predicción de movimiento entre vistas, una PU actual 134 puede corresponder a un área de referencia con el mismo tamaño que la PU actual identificada por el vector de disparidad en una vista de referencia (V0) 136. En algunos casos, el área de referencia puede tener información de movimiento más rica que la necesaria para la generación de solo un conjunto de información de movimiento para la PU actual 134. Por lo tanto, se propone un procedimiento de predicción de movimiento entre vistas en el nivel de sub-PU (SPIVMP) en el que la actual PU 134 se divide en múltiples sub-PU, y el vector de disparidad (DV) se usa para identificar los bloques de referencia 138 en la vista de referencia (V0) 136 para cada una de las sub-PU de la PU actual 134. En la modalidad de SPIVMP, cada una de las sub-PU de la PU actual 134 tiene un conjunto completo de información de movimiento copiada de los bloques de referencia 138, de modo que la PU actual 134 pueda contener múltiples conjuntos de información de movimiento. La modalidad de SPIVMP también puede ser señalizada como una candidata especial de modalidad de fusión, de manera similar a la modalidad de BVSP de sub-PU descrita anteriormente.

[0125] En la solicitud provisional estadounidense N.º 61/858.089, presentada el 24 de julio de 2013, titulada "PREDICCIÓN DE MOVIMIENTO AVANZADA PARA LA CODIFICACIÓN DE VÍDEO TRIDIMENSIONAL", para Ying Chen y Li Zhang, se propone que el candidato a la MPI también se pueda extender de una manera similar a la predicción de movimiento entre vistas en el nivel de sub-PU. Por ejemplo, si la PU de profundidad actual tiene una región cosituada que contiene múltiples PU, la PU de profundidad actual se puede separar en sub-PU, cada una de las cuales puede tener un conjunto diferente de información de movimiento. Este procedimiento puede denominarse MPI de sub-PU.

[0126] Los diseños de sub-PU en la norma 3D-HEVC, incluyendo la BVSP de sub-PU, la predicción de movimiento entre vistas de sub-PU y la MPI de sub-PU, descritas anteriormente, pueden experimentar algunos problemas.

[0127] Primero, como se ha descrito anteriormente, en la modalidad de BVSP de sub-PU, la compensación de movimiento se realiza para predecir cada una de las sub-PU de una PU actual basándose en información independiente de movimiento de sub-PU obtenida, basándose en los correspondientes bloques de profundidad de una imagen de referencia entre vistas, identificada por un vector de movimiento de disparidad seleccionado para la PU actual. Sin embargo, después de realizar la compensación de movimiento para predecir cada una de las sub-PU, solo se almacena el vector de disparidad de movimiento correspondiente a cada lista de imágenes de referencia, para la PU actual. En este caso, cuando se usa la PU actual para predecir una PU posterior, se considera que las sub-PU de la PU actual tienen los mismos vectores de movimiento, de modo que la modalidad de BVSP de sub-PU tenga poco impacto en la mejora de la precisión de la predicción del vector de movimiento.

[0128] Segundo, el diseño actual de sub-PU solo se habilita cuando la predicción entre capas está habilitada. Sin embargo, el diseño de sub-PU puede ser aplicable para mejorar la precisión del uso de la compensación de movimiento en la predicción de capa única. Además, el diseño actual de sub-PU solo es aplicable a la modalidad de interpredicción de fusión, en el que no se permite ningún refinamiento adicional de los vectores de movimiento candidatos. El diseño actual de sub-PU, por lo tanto, no se puede habilitar para las modalidades de predicción de movimiento donde se permite el refinamiento de movimiento, por ejemplo, la modalidad de AMVP para códecs basados en la HEVC.

[0129] En tercer lugar, los filtros de desbloqueo utilizados en la HEVC habitualmente se mantienen sin cambios en las extensiones de la HEVC, incluyendo la 3D-HEVC. Sin embargo, es posible que el diseño actual del filtro de desbloqueo en la HEVC no filtre las fronteras de las sub-PU, ya que se supone en la HEVC que los bloques en la misma TU dentro de una PU se compensan en movimiento, como un bloque completo de una sola imagen. En este

caso, no se espera que haya distorsiones de pixelado presentes o eliminadas desde dentro de una PU. Para desbloquear las fronteras de sub-PU sin cambiar el diseño del filtro de desbloqueo en la extensión 3D-HEVC, esta divulgación describe la conversión de una sub-PU en una estructura favorecedora del desbloqueo antes de que se aplique un proceso de filtro de desbloqueo. Por ejemplo, antes del proceso de filtrado de desbloqueo, cada PU que utiliza el diseño de sub-PU puede convertirse en uno o más árboles de codificación, donde cada sub-PU puede convertirse en una PU en una CU que es un nodo del respectivo árbol de codificación. En este ejemplo, la estructura del árbol de codificación es compatible con el diseño sintáctico de la HEVC.

[0130] Sin embargo, algunos problemas adicionales pueden ocurrir con la solución anterior. Como primer ejemplo, si una unidad de predicción A (PU A) se ha codificado con interpredicción normal, y otra unidad de predicción (PU B) dentro de la misma CU que la PU A se codifica con sub-PU, será necesario que las PU se conviertan en dos CU y el filtrado puede ocurrir dentro de la PU A, que no está subdividida. Como segundo ejemplo, cuando una TU vale para una CU completa, que incluye al menos una PU con sub-PU, la CU puede convertirse en múltiples CU, la HEVC no da soporte a múltiples CU con la misma TU.

[0131] Esta divulgación describe varias técnicas para abordar los problemas descritos anteriormente.

[0132] Con respecto al primer problema descrito anteriormente, en la modalidad de BVSP de sub-PU en la norma 3D-HEVC, después de realizar la compensación de movimiento para predecir cada una de las sub-PU de la PU actual en función de información distinta de movimiento de sub-PU, obtenida basándose en correspondiente bloques de profundidad de una imagen de referencia entre vistas, identificada por un vector de movimiento de disparidad seleccionado para la PU actual, solo el vector de movimiento de disparidad correspondiente a cada lista de imágenes de referencia se almacena para la PU actual. El único vector de movimiento de disparidad se almacena para cada lista de imágenes de referencia, incluso aunque la compensación de movimiento de la PU se basa en múltiples vectores de movimiento incluidos en la información de movimiento distinta, obtenida para las múltiples sub-PU de la PU actual.

[0133] Según las técnicas de esta divulgación, para cada PU predicha utilizando la modalidad de BVSP de sub-PU, un dispositivo de codificación de vídeo, es decir, el codificador de vídeo 20 y/o el decodificador de vídeo 30, preserva, por ejemplo, almacena o mantiene, la información de movimiento distinta obtenida para cada una de las sub-PU de la PU actual incluso después de que se realiza la compensación de movimiento. Como ejemplo, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden almacenar el vector de movimiento de disparidad obtenido para cada una de las sub-PU en una memoria, tal como un almacén temporal de imágenes decodificadas, con una lista de imágenes de referencia que incluye la imagen de referencia entre vistas, identificada por un índice de vista de referencia asociado al vector de movimiento de disparidad para la PU actual.

[0134] La información de movimiento adicional, más rica, almacenada para la PU actual se puede usar entonces para predecir las PU posteriores para las cuales la PU actual es un bloque vecino. Por ejemplo, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden generar una lista de candidatas, en modalidad de fusión, de información de movimiento, para predecir una PU posterior que incluye la información de movimiento almacenada para al menos una de las sub-PU de la PU actual, como una candidata de BVSP de sub-PU en la lista de candidatas en modalidad de fusión. En un ejemplo, si la PU posterior se codifica en la modalidad de BVSP de sub-PU, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden no necesitar obtener la información de movimiento distinta para cada una de las sub-PU de la PU posterior. En cambio, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden, en cambio, seleccionar la información de movimiento distinta como la candidata de BVSP de sub-PU de la lista de candidatas, en modalidad de fusión, de información de movimiento para predecir las sub-PU de la PU posterior. La operación de predecir una PU en la modalidad de BVSP de sub-PU y almacenar la información de movimiento obtenida para cada una de las sub-PU de la PU se describe con más detalle a continuación con respecto a la figura 17.

[0135] Con respecto al segundo problema descrito anteriormente, esta divulgación describe técnicas para realizar una modalidad de TMVP avanzada para predecir las sub-PU de una PU en la codificación de una sola capa para la cual se puede permitir el refinamiento del vector de movimiento. El codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden configurarse para realizar la modalidad de TMVP avanzada, que incluye la determinación de vectores de movimiento para la PU en al menos dos etapas, para obtener información de movimiento para la PU que incluye diferentes vectores de movimiento e índices de referencia para cada una de las sub-PU de la PU.

[0136] Las figuras 14A y 14B son diagramas conceptuales que ilustran la modalidad de TMVP avanzada para predecir las sub-PU en una PU en codificación de capa única. Como se ilustra en la figura 14A, la primera etapa 170 de la modalidad de TMVP avanzada determina un vector de movimiento que identifica un correspondiente bloque 174 en una imagen de referencia para la PU actual 173 en la imagen actual, y la segunda etapa 172 de la modalidad de TMVP avanzada extrae múltiples conjuntos de información de movimiento desde las sub-PU del correspondiente bloque 174 en la imagen de referencia y asigna cada uno de los conjuntos de información de movimiento a una de las sub-PU de la PU actual 173 en la imagen actual. Cada sub-PU de la PU actual 173, por lo tanto, se compensa en movimiento por separado. El vector de movimiento en la primera etapa 170 puede obtenerse de bloques vecinos espaciales y temporales de la PU actual 173. En un ejemplo, el vector de movimiento de la primera etapa puede seleccionarse como candidato de modalidad de fusión entre todos los demás candidatos de modalidad de fusión, que puede incluir o no un candidato similar al candidato de TMVP en la HEVC. En otro ejemplo, el vector de movimiento

de la primera etapa se puede seleccionar como candidato de la modalidad de AMVP entre todos los demás candidatos de la modalidad de AMVP y se puede refinar. En este ejemplo, cuando una sub-PU encuentra un vector de movimiento no disponible para cada dirección de predicción, se puede usar un vector de movimiento representativo.

[0137] Aplicable a la codificación de una sola capa y a la TMVP de sub-PU, el codificador de vídeo 20 o el decodificador de vídeo 30 pueden determinar datos de refinamiento de movimiento, por ejemplo, una diferencia de vectores de movimiento, para una PU o CU que se señaliza con los índices de predictor de movimiento para las listas de candidatos de MV. En un ejemplo, para cada dirección de predicción, se puede determinar una única diferencia de vectores de movimiento, y es aplicable a todos los vectores de movimiento de las sub-PU o PU. En otro ejemplo, para cada dirección de predicción, se pueden determinar diferencias distintas de vectores de movimiento para cada una de las sub-PU o PU. Además, para cada componente horizontal o vertical, los valores de diferencias de vectores de movimiento pueden transformarse, y los coeficientes transformados resultantes pueden cuantizarse o truncarse adicionalmente y codificarse de manera similar a los residuos de píxeles en los códecs de vídeo.

[0138] En otro ejemplo, similar a la HEVC, los datos de refinamiento de movimiento para los vectores de movimiento de sub-PU pueden transmitirse desde el codificador de vídeo 20 al decodificador de vídeo 30 cuando un candidato de MV de sub-PU se agrega a una lista de candidatos en modalidad de AMVP y no se agrega a una lista de candidatos en modalidad de fusión. En un ejemplo alternativo, un candidato de MV de sub-PU puede valer solo si los valores de índices de referencia asociados a todos los vectores de movimiento de las sub-PU o PU son los mismos. En otro ejemplo alternativo, un candidato de MV de sub-PU puede valer siempre y los valores de índices de referencia asociados a todos los vectores de movimiento de las sub-PU se transmiten explícitamente. Además, si se aplica la cuantización o la transformación de los valores de diferencias de vectores de movimiento, los vectores de movimiento pueden ajustarse a escala hacia una imagen de referencia fija. Una vez que se recopilan las diferencias de vectores de movimiento, las diferencias se agregan a los vectores de movimiento ajustados a escala. Posteriormente, los nuevos vectores de movimiento se reducen a escala hacia sus respectivas imágenes de referencia diferentes, identificadas por los diferentes valores de índices de referencia de las sub-PU o las PU.

[0139] La siguiente sección proporciona detalles ejemplares de implementación con respecto a la modalidad de TMVP avanzada, descrita anteriormente, para casos donde los valores de índices de referencia asociados a los vectores de movimiento de las sub-PU o las PU son diferentes. La identificación del vector de movimiento de la primera etapa será expuesta primero. El vector de movimiento de la primera etapa se convierte a partir de los bloques adyacentes espaciales de la PU actual, que contienen vectores de movimiento temporal. Los bloques adyacentes espaciales pertenecen a los utilizados para la predicción típica de vectores de movimiento, por ejemplo, los bloques utilizados en la AMVP y la fusión para la HEVC.

[0140] Cuando los bloques adyacentes espaciales tienen vectores de movimiento candidatos que están asociados a diferentes valores de índices de referencia, se aplica uno de los siguientes procesos de decisión para decidir qué índice de referencia se usa para identificar la imagen de referencia a partir de la cual se determina la información de movimiento a nivel de sub-PU.

1. Se eligen los bloques con el valor de índice de referencia más pequeño. Entre ellos, aquel al que se accede anteriormente se elige para devolver el vector de movimiento temporal que sea el "vector de la primera etapa". Se supone que se accede a los bloques en un orden determinado en función de las ubicaciones espaciales relativas de los bloques.

2. Se eligen los bloques con el valor de índice de referencia más pequeño. Los vectores de movimiento de estos bloques se promedian (si son más de uno) para ser el "vector de la primera etapa"; alternativamente, se puede usar una operación de media.

3. Se elige el índice de referencia con una frecuencia máxima entre los índices de referencia de los bloques vecinos. Entre los bloques que tienen ese índice de referencia, se elige el vector de movimiento al que se accede primero, o se usa el promedio (por ejemplo, la media) de los vectores de movimiento para ser el "vector de la primera etapa". Alternativamente, se pueden aplicar otras funciones matemáticas para obtener el vector de movimiento de la primera etapa.

4. Se elige el índice de referencia con la distancia de POC más cercana a la imagen actual. Entre los bloques que tienen ese índice de referencia, se elige el vector de movimiento al que se accede primero, o se usa un promedio (por ejemplo, la media u otra función matemática) de los vectores de movimiento para ser el "vector de la primera etapa".

5. Un índice de referencia primario se señaliza en el encabezado de fragmento y se eligen bloques con un índice de referencia igual al índice de referencia primaria para producir el "vector de la primera etapa", con procedimientos similares a los descritos anteriormente. Cuando ningún bloque tiene un índice de referencia igual al índice de referencia primaria, se pueden usar los procedimientos descritos anteriormente para elegir el índice de referencia.

6. Dado que la imagen utilizada para la TMVP suele ser fija para cada fragmento, como en la AVC y la HEVC, el

índice de referencia primaria puede ser el mismo que el índice de referencia que indica la TMVP.

7. El índice de referencia anterior podría ser un índice para la ListaImgRef0 o la ListaImgRef1. Alternativamente, una unión de listas de imágenes de referencia (UListasImgRef) puede ser generada por la ListaImgRef0 y la ListaImgRef1, y el índice de referencia puede ser el índice para la UListasImgRef. Téngase en cuenta que cualquier imagen identificada por la UListasImgRef pertenece a ListaImgRef0 o a ListaImgRef1, o a ambas, y no hay ninguna imagen que pertenezca a la UListasImgRef pero que no esté en la ListaImgRef0 o la ListaImgRef1. La UListasImgRef no tiene dos imágenes idénticas. Alternativamente, y además, la UListasImgRef solo puede contener imágenes de referencia temporales dentro de la misma capa, o marcadas como imágenes a corto plazo.

8. Alternativamente, el índice de referencia y el "vector de la primera etapa" se pueden seleccionar entre los candidatos en modalidad de fusión.

a. En un ejemplo, el índice de referencia y el "vector de la primera etapa" se seleccionan a partir de un candidato espacial en modalidad de fusión, obtenido de una posición de bloque relativa, por ejemplo, bloque vecino izquierdo.

b. Alternativamente, el índice de referencia y el "vector de la primera etapa" pueden seleccionarse del primer candidato disponible en la lista de candidatos en modalidad de fusión.

c. Además, cuando el candidato de fusión seleccionado utiliza la predicción doble, el vector de movimiento y el índice de referencia pueden seleccionarse a partir de uno de los conjuntos fusionados de información de movimiento.

[0141] Cuando se identifican un índice de referencia y el "vector de la primera etapa", se identifica la imagen de referencia que se utiliza para determinar la información de movimiento de sub-PU, así como la región en la imagen de referencia correspondiente a la PU actual. En un caso, un índice de referencia puede indicar una imagen de referencia que es diferente a la imagen a utilizar para la TMVP debido, por ejemplo, a que el índice de referencia se obtiene y la imagen utilizada para la TMVP se señaliza explícitamente. En este caso, el índice de referencia se puede cambiar para identificar la imagen utilizada para la TMVP, y el vector de movimiento se puede ajustar a escala hacia la imagen utilizada para la TMVP en función de las distancias de POC.

[0142] En la descripción anterior, la identificación del "vector de la primera etapa" solo utiliza bloques adyacentes espaciales. Alternativamente, los bloques vecinos temporales se pueden usar para identificar el "vector de la primera etapa", donde las posiciones de los bloques vecinos temporales son similares a los bloques usados en los NBDV. Dichas posiciones incluyen una posición central de la PU actual o una posición inferior derecha de la PU actual, cada una de las cuales se encuentra en una imagen de referencia identificada.

[0143] Ahora se expondrá la generación de movimiento de sub-PU para la TMVP. Cada sub-PU de una PU actual puede ubicar una sub-PU correspondiente dentro del bloque correspondiente de la imagen de referencia identificada por el "vector de primera etapa". Se genera un conjunto de información de movimiento obtenida a partir de la información de movimiento dentro de la sub-PU correspondiente de la imagen de referencia, para cada una de las sub-PU en la PU actual, de manera similar a la predicción de movimiento entre vistas en el nivel de sub-PU, descrita anteriormente. Si la información de movimiento de las sub-PU correspondientes no está disponible, el "vector de la primera etapa" se puede usar para las sub-PU de la PU actual como sustituto. Dado que cada sub-PU puede tener un índice de referencia diferente para cada dirección de predicción, se proponen varias técnicas para ajustar a escala los vectores de movimiento hasta una imagen de referencia de destino, para cada lista de imágenes de referencia.

[0144] Como se ilustra en la figura 14B, la imagen de referencia 183 identificada por el vector de movimiento 186 dentro de la correspondiente sub-PU 185 de la imagen de referencia del origen de movimiento 182 es ImgOri, la imagen de referencia (es decir, la imagen del origen del movimiento) 182 que contiene la correspondiente sub-PU 185 es ImgT, la imagen actual 180 es ImgAct y la imagen de referencia de destino 184 es ImgDest. Se supone, para los fines de la exposición, que el vector de movimiento 186 es MV, y el vector de movimiento ajustado a escala es MV' (es decir, el predictor de vector de movimiento 188 para predecir una sub-PU de la PU 181 en la imagen actual 180). Téngase en cuenta que cuando un vector de movimiento temporal 187, identificado en la primera etapa de la modalidad de TMVP avanzada, se asocia a una imagen de referencia que no es la imagen a partir de la cual se obtiene la información de movimiento de sub-PU, el ajuste a escala de los vectores de movimiento en función de la distancia de POC puede ser también posible.

1. Ajustar a escala el vector de movimiento hacia un índice de referencia fijo de la imagen actual:

$$MV' = MV * (POC(ImgDest) - POC(ImgAct))/(POC(ImgOri) - POC(ImgT)),$$

donde la función POC () devuelve el valor de POC de una imagen dada. Téngase en cuenta que las operaciones anteriores de multiplicación y división se pueden simplificar de una manera similar a la de la TMVP de la HEVC.

a. El índice de referencia fijo puede ser el mismo para todo el fragmento, por ejemplo, puede fijarse igual a 0 como en la HEVC.

b. El índice de referencia fijo puede obtenerse de los bloques adyacentes espaciales.

2. Ajustar a escala el vector de movimiento siempre hacia la imagen de referencia de la sub-PU correspondiente, que es ImgOri: MV' = MV * (POC (ImgOri) - POC (ImgAct))/(POC (ImgOri) - POC (ImgT)).

3. Ajustar a escala el vector de movimiento siempre hacia la imagen de ubicación conjunta, que es ImgT:

MV' = MV * (POC (ImgT) - POC (ImgAct))/(POC (ImgOri) - POC (ImgT)).

[0145] Como se ha indicado anteriormente, la imagen de referencia de destino de cada lista de imágenes de referencia se fija en la imagen de referencia que tiene un índice de referencia igual a 0 en la lista de imágenes de referencia. En otro ejemplo, la imagen de referencia de destino de cada lista de imágenes de referencia se fija en la misma imagen de referencia identificada junto con el "vector de la primera etapa", como se ha descrito anteriormente. El funcionamiento de la TMVP avanzada para la predicción del vector de movimiento de nivel sub-PU se describe con más detalle a continuación con respecto a la figura 19.

[0146] Con respecto al tercer problema descrito anteriormente, esta divulgación describe técnicas relacionadas con los procesos de filtro de desbloqueo aplicados a las CU de un bloque de vídeo que incluyen al menos una PU con múltiples sub-PU. Las técnicas de esta divulgación permiten que las fronteras de sub-PU se desbloqueen convirtiendo las sub-PU en una estructura favorecedora del desbloqueo, de modo que los filtros de desbloqueo de la HEVC puedan seguir utilizándose para los bloques de la norma 3D-HEVC. De esta manera, la entrada al filtro de desbloqueo puede cambiar para permitir el filtrado de las distorsiones en bloques a lo largo de las fronteras de las sub-PU, mientras se mantiene el filtro de desbloqueo sin cambios. En esta divulgación, una CU que está al menos parcialmente codificada con las sub-PU se menciona como una CU avanzada, y una PU codificada con las sub-PU dentro de una CU avanzada codificada se refiere a una PU avanzada. Otras PU en la CU avanzada, si las hay, se denominan PU normales.

[0147] En esta divulgación se describen varias técnicas para introducir los bordes de las sub-PU al filtro de desbloqueo, para eliminar las distorsiones a lo largo de las fronteras de sub-PU. Un proceso de conversión de sub-PU, como se describe en esta divulgación, puede introducirse en la 3D-HEVC justo antes de un proceso de filtrado de desbloqueo, es decir, después de reconstruir un bloque de vídeo para almacenarlo como una imagen de referencia pero antes de almacenar realmente el bloque de vídeo reconstruido en un almacén temporal de imágenes decodificadas del codificador de vídeo 20 y/o del decodificador de vídeo 30.

[0148] En un primer ejemplo, justo antes de aplicar un filtro de desbloqueo a la CU que incluye las PU con las sub-PU, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden configurarse para convertir cada PU que utiliza el diseño de sub-PU en uno o más árboles de codificación, de modo que cada una de las sub-PU pueda convertirse en una PU en las unidades de codificación que son los nodos de los árboles de codificación. En este ejemplo, las fronteras de sub-PU se convierten en fronteras de PU artificiales para los fines del filtro de desbloqueo. La estructura del árbol de codificación es preferentemente compatible con el diseño sintáctico de la HEVC.

[0149] En otros ejemplos, antes de aplicar un filtro de desbloqueo a una CU del bloque de vídeo que incluye la PU con las sub-PU, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden configurarse, en cambio, para convertir la CU a fin de crear fronteras artificiales de PU o fronteras artificiales de TU en las fronteras de sub-PU. En un ejemplo, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden configurarse para convertir un árbol de transformación de la CU a fin de asociar la PU con una jerarquía de árbol de transformación y asociar cada una de las sub-PU a una TU. En este ejemplo, las fronteras de sub-PU se convierten en fronteras de TU artificiales para los fines del filtro de desbloqueo. En otro ejemplo, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden configurarse para convertir la CU en un árbol de codificación a fin de asociar la PU a una CU y asociar cada una de las sub-PU a una PU. En este ejemplo, las fronteras de sub-PU se convierten en fronteras de PU artificiales para los fines del filtro de desbloqueo. La estructura del árbol de codificación es preferentemente compatible con el diseño sintáctico de la HEVC.

[0150] En cualquiera de los ejemplos anteriores, después de convertir las sub-PU en estructuras favorecedoras del desbloqueo, el filtro de desbloqueo puede aplicarse a las fronteras de la PU entre dos PU adyacentes de la CU y/o a las fronteras de la TU entre dos TU adyacentes de la CU, incluidas las fronteras de PU artificiales y las fronteras de TU artificiales.

[0151] Para el ejemplo descrito anteriormente en el que el árbol de transformación de la CU se convierte a fin de asociar la PU a una jerarquía de árbol de transformación y asociar cada una de las sub-PU a una TU, de modo que las fronteras de la sub-PU se conviertan en fronteras de TU artificiales, se pueden aplicar uno o más de los siguientes procesos de decisión.

1. Cuando el árbol de transformación tiene una raíz con el indicador_transformación_dividida igual a 1, los nodos

correspondientes a la PU normal (si acaso) de la CU avanzada no se cambian.

2. Cuando el árbol de transformación tiene una raíz con un indicador de transformación dividida igual a 1, se cambia un nodo en una PU avanzada para introducir una jerarquía de árbol de transformación, de la siguiente manera:

a. Para el nodo actual, si el indicador de transformación dividida es 0, y el tamaño de TU es mayor que el tamaño de la sub-PU, fijar el indicador de transformación dividida en 1 y, para cada uno de los cuatro nodos secundarios, se aplica lo siguiente:

i. Fijar cbf_luma, cbf_cb y cbf_cr del nodo para que sean iguales a los del nodo principal y el indicador_transformación_dividida igual a 0;

ii. Fijar el nodo actual en el nodo secundario e ir a la etapa a.

b. Para el nodo actual, si el indicador de transformación dividida es 1, para cada uno de los cuatro nodos secundarios, se aplica lo siguiente: fijar el nodo actual en el nodo secundario e ir a la etapa a.

3. Alternativamente, cuando se introduce una jerarquía de árbol de transformación para una PU y cbf_luma, cbf_cb y cbf_cr se fijan en un nodo recién dividido, los cbf_luma, cbf_cb y cbf_cr del nodo se fijan en 0.

4. Alternativamente, cuando se introduce una jerarquía de árbol de transformación para una PU y cbf_luma (o cbf_cb, o cbf_cr) se fijan en un nodo recién dividido, cbf_luma (o cbf_cb, o cbf_cr) del nodo se fija en cualquier valor entero positivo no nulo (por ejemplo, 1) si cbf_luma (o cbf_cb, o cbf_cr) no es igual a 0, o 0 en caso contrario.

5. Cuando el árbol de transformación tiene una raíz con un indicador de transformación dividida igual a 0, una PU normal (si la hubiera) se divide en unidades de transformación, con las siguientes etapas en orden.

a. El indicador_transformación_dividida de la CU se fija primero en 1.

b. Si la partición de la CU es de tamaño NxN, la PU normal corresponde a un nodo. Si la partición de la CU es de tamaño 2NxN o Nx2N, la PU normal corresponde a dos nodos.

c. Cada uno de los nodos anteriores dentro de una PU normal está configurado para tener el indicador_transformación_dividida fijado en 0 y cbf_luma, cbf_cb y cbf_cr fijados iguales a cbf_luma, cbf_cb y cbf_cr de la unidad de transformación original que abarca toda la CU avanzada.

d. Para cada una de las PU avanzadas de la CU, primero se generan 1 o 2 nodos que abarcan la región cuadrada de la PU y, para cada nodo, la jerarquía del árbol de transformación se introduce de manera similar a la de la etapa 2, 3 o 4 anterior.

6. Como alternativa a la etapa 5 anterior, para cada uno de los nodos anteriores dentro de una PU normal, cbf_luma, cbf_cb y cbf_cr se fijan iguales a 0.

7. Como alternativa a la etapa 4 anterior, para cada uno de los nodos anteriores dentro de una PU normal, su cbf_luma (o cbf_cb o cbf_cr) se fija igual a cualquier valor entero positivo no nulo (por ejemplo, 1) si cbf_luma (o cbf_cb o cbf_cb) no es igual a 0, o 0 de lo contrario.

[0152] Para el ejemplo descrito anteriormente en el que la CU se convierte en un árbol de codificación para asociar la PU a una CU y asociar cada una de las sub-PU a una PU de tal manera que las fronteras de sub-PU se conviertan en fronteras de PU artificiales, uno o más de los siguientes procesos de decisiones pueden ser aplicados.

1. Después de la conversión, la CU actual avanzada se convertirá en la raíz del árbol de codificación, en concreto, la raíz convertida, que es un árbol cuádruple que contiene al menos cuatro unidades de codificación.

2. Para cualquier PU normal en una CU avanzada, se fija en una o más unidades de codificación.

3. Además, el valor de CBF de cada componente de las unidades de codificación convertidas, convertidas a partir de las PU normales en una CU avanzada, se fija además en 0. De esta manera, se evita el filtrado dentro de una PU de interpredicción normal, incluso si la PU se ha convertido en dos CU.

a. Para cualquier PU avanzada, el valor de CBF del árbol de codificación o de la unidad de codificación convertidos, que está un nivel por debajo de la raíz convertida, se fija como no nulo si el valor de CBF de la CU avanzada actual es no nulo.

b. Además, para cualquier CU o PU o árbol de codificación convertidos, dentro de una PU avanzada, si originalmente no tiene un valor de CBF señalizado, se establece que sea el mismo que el valor de CBF de su

nodo padre en el árbol cuádruple, o el mismo estado nulo o no nulo que el valor de CBF de su nodo padre en el árbol cuádruple.

4. Alternativamente, para una CU avanzada, no se puede aplicar una unidad de transformación única para la CU completa, lo que significa que si una PU normal está dentro de la CU avanzada, debe contener una unidad de transformación que no sea compartida por otra PU de la CU.

a. Alternativamente, o además, para cada PU avanzada, los nodos de hoja del árbol de transformación se distribuyen un nivel más alto que la sub-PU o más profundo, lo que significa que cada uno de las cuatro sub-PU dentro de una CU convertida tiene señalizada una unidad de transformación única.

5. Más específicamente, si una CU con un tamaño de 2Lx2L contiene una PU que utiliza el diseño de sub-PU, entonces se realiza la siguiente conversión para todas las PU en la CU. Se fija un indicador de división igual a 1 para la CU avanzada actual y se aplica lo siguiente y el nodo actual se configura para que sea la CU:

a. Para cada una de las áreas cuadradas de un cuarto del nodo actual en el orden de exploración ráster, se aplica lo siguiente:

i. Establecer este área de un cuarto como nodo secundario.

ii. Si el nodo secundario no está codificado con las sub-PU, se fija que sea una unidad de codificación (con el indicador de división igual a 0) con partición de tamaño 2Nx2N.

1. Alternativamente, o además, el valor de CBF de cada componente del nodo secundario se fija en 0.

2. Como alternativa, o además, el indicador de división de TU de la unidad de codificación se fija en 0.

iii. De lo contrario, si el nodo secundario está codificado con las sub-PU y contiene más de 1 sub-PU, un indicador de división se fija en 1 para el nodo secundario (considerado por tanto como un árbol de codificación) y se aplica lo siguiente:

1. Como alternativa, además, si el valor de CBF no está presente para el nodo secundario, se establece que sea igual al valor de CBF del nodo actual del nivel superior (el nodo padre de este nodo secundario).

2. Establecer el nodo secundario como el nodo actual e ir a la etapa a.

iv. De lo contrario, si el nodo secundario solo contiene 1 sub-PU, se establece que el nodo secundario sea la hoja del árbol de codificación, por tanto, una unidad de codificación (con el indicador de división igual a 0).

1. Se establece que la modalidad de partición del nodo secundario sea la modalidad de partición de la sub-PU

a. Si la modalidad de partición de la sub-PU es de tamaño 2Nx2N, la sub-PU contiene un bloque de tamaño 2Nx2N y se establece que la modalidad de partición de la unidad de codificación sea de tamaño 2Nx2N.

b. Si la modalidad de partición de la sub-PU es de tamaño Nx2N, la sub-PU contiene dos bloques de tamaño Nx2N y se establece que la modalidad de partición de la unidad de codificación sea de tamaño Nx2N.

c. Si la modalidad de partición de la sub-PU es de tamaño 2NxN, la sub-PU contiene dos bloques de tamaño 2NxN y se establece que la modalidad de partición de la unidad de codificación sea de tamaño 2NxN.

2. Alternativamente, o además, si el valor de CBF no está presente para el nodo secundario, se establece que sea igual al valor de CBF del nodo actual de nivel más alto (el nodo padre de este nodo secundario).

3. Como alternativa, o además, el indicador de división de TU de la unidad de codificación se fija en 0.

[0153] Como otra alternativa para el ejemplo descrito anteriormente en el que el árbol de transformación de la CU se convierte a fin de asociar la PU a una jerarquía de árbol de transformación y asociar cada una de las sub-PU a una TU de tal manera que las fronteras de sub-PU se conviertan en fronteras de TU artificiales, se pueden aplicar uno o más de los siguientes procesos de decisión para al menos una parte de cada sub-PU.

1. Cuando el árbol de transformación tiene una raíz con el indicador_transformación_dividida igual a 1, los nodos normales que contienen solo las PU normales (o parte de las PU normales) de la CU avanzada no se cambian.

2. Cuando el árbol de transformación tiene una raíz con el indicador_transformación_dividida igual a 1, un nodo avanzado que contiene cualquier sub-PU (o parte de sub-PU) se cambia para introducir la jerarquía del árbol de transformación de la siguiente manera:

a. Para el nodo actual, si contiene cualquier sub-PU (o parte de sub-PU), y el indicador_transformación_dividida es 0, y el tamaño de TU es mayor que el tamaño de sub-PU, fijar el indicador_transformación_dividida en 1, y para cada uno de los cuatro nodos secundarios, se aplica lo siguiente:

i. Establecer cbf luma, cbf_cb y cbf_cr del nodo para que sean iguales a los del nodo padre y el indicador_transformación_dividida igual a 0;

ii. Fijar el nodo actual en el nodo secundario e ir a la etapa a.

b. Para el nodo actual, si indicador_transformación_dividida es 1, para cada uno de los cuatro nodos secundarios, se aplica lo siguiente: fijar el nodo actual en el nodo secundario e ir a la etapa a.

3. Alternativamente, cuando se introduce una jerarquía de árbol de transformación para un nodo y cbf_luma, cbf_cb y cbf_cr se establecen en un nodo recién dividido, cbf_luma, cbf_cb y cbf_cr del nodo recién dividido se fijan en 0.

4. Alternativamente, cuando se introduce la jerarquía de árbol de transformación para un nodo y cbf_luma, cbf_cb y cbf_cr se establecen en un nodo recién dividido, cbf_luma (o cbf_cb, o cbf_cr) del nodo recién dividido se fija en cualquier valor entero positivo no nulo (por ejemplo, 1) si cbf_luma (o cbf_cb, o cbf_cr) no es igual a 0, o 0 de lo contrario.

5. Cuando el árbol de transformación tiene una raíz con indicador_transformación_dividida igual a 0, un nodo normal que contiene solo las PU normales (o parte de las PU normales) debe dividirse en unidades de transformación, con las siguientes etapas en orden.

a. El indicador_transformación_dividida de la CU se fija primero en 1.

b. Si la partición de la CU es de tamaño NxN, la PU normal corresponde a un nodo normal. Si la partición de la CU es de tamaño 2NxN o Nx2N, la PU normal corresponde a dos nodos normales. Si la partición de la CU es de tamaño 2NxnU, 2NxnD, nLx2N o nRx2N, la PU normal corresponde a 0 o 2 nodos normales.

c. Cada uno de los nodos normales anteriores se establece para que tenga el indicador_transformación_dividida fijado en 0 y cbf_luma, cbf_cb, y cbf_cr fijados iguales a cbf_luma, cbf_cb, y cbf_cr de la unidad de transformación original que abarca la totalidad de la CU avanzada.

d. Para cada uno de los nodos avanzados que contiene cualquier sub-PU (o parte de sub-PU) de la CU, la jerarquía del árbol de transformación se introduce de manera similar a como se indica en las etapas 2, 3 o 4 anteriores.

6. Como alternativa al paso 5 anterior, para cada uno de los nodos normales anteriores, cbf_luma, cbf_cb y cbf_cr se fijan iguales a 0.

7. Alternativamente a la etapa 4 anterior, para cada uno de los nodos normales anteriores, su cbf_luma (o cbf_cb o cbf_cr) se establece igual a cualquier valor entero positivo no nulo (por ejemplo, 1) si cbf_luma (o cbf_cb o cbf_cb) no es igual a 0, o 0 en caso contrario.

[0154] Algunos ejemplos de detalles de implementación se proporcionan a continuación. El texto a continuación indica las modificaciones propuestas al borrador WD1 de la norma 3D-HEVC, para la implementación de algunas de las técnicas descritas en esta divulgación.

PROCESO DE DECODIFICACIÓN

H.8.1 Proceso de decodificación general

[0155] 3. Los procesos en las subcláusulas 8,4; 8,5; 8,6; 8,7 y 8,8 especifican procesos de decodificación utilizando elementos sintácticos en todas las capas de estructura sintáctica. Es un requisito de conformidad del flujo de bits que los fragmentos codificados de la imagen contengan datos de segmentos de fragmentos para cada unidad de árbol de codificación de la imagen, de modo que la división de la imagen en fragmentos, la división de los fragmentos en segmentos de fragmentos y la división de los segmentos de fragmentos en unidades de árbol de codificación formen, cada una, una partición de la imagen.

H.8.5.3.3.2 Proceso de selección de imágenes de referencia

[0156] La entrada a este proceso es un índice de referencia idxrefLX.

[0157] El resultado de este proceso es una imagen de referencia que consiste en una formación bidimensional de muestras de luma $imgRefLX_L$ y dos formaciones bidimensionales de muestras de croma $imgRefLX_{Cb}$ y $imgRefLX_{Cr}$.

[0158] La imagen de referencia de salida ListaImgRefX [idxrefLX] consta de una formación, de tamaño ancho de imagen en muestras de luma por altura de imagen en muestras de luma, de muestras de luma $imgRefLX_L$ y dos formaciones, de tamaño AnchoImgEnMuestrasC por AlturaImgEnMuestrasC, de muestras de croma $imgRefLX_{Cb}$ y $imgRefLX_{Cr}$.

[0159] Las formaciones de muestras de imágenes de referencia $imgRefLX_L$, $imgRefLX_{Cb}$ e $imgRefLX_{Cr}$ corresponden a las formaciones de muestras decodificadas S_L, S_{Cb} y S_{Cr} obtenidas por la subcláusula 8.7 y la subcláusula 8.8 para una imagen decodificada previamente.

H.8.7 Proceso de modificación del árbol de transformación

H.8.7.1 General

[0160] Las entradas a este proceso son: formación de indicadores de división del árbol de codificación indicador_cu_dividida, formación de modalidades de partición de la unidad de codificación ModalidadPart, indicador de división del árbol de transformación indicador_transformación_dividida, formación de indicadores de subunidad de predicción de la unidad de codificación indicadorSubPu, tamaño de la subunidad de predicción tamañoSubPu. El resultado de este proceso es la formación modificada de indicadores de división del árbol de transformación indicador_transformación_dividida.

[0161] Para cada unidad de codificación, si se codifica en la modalidad de interpredicción y contiene una unidad de predicción que utiliza un diseño de subunidad de predicción, el indicador de división del árbol de transformación se modifica para hacer que la frontera de subunidad de predicción secundaria sea frontera de unidad de transformación:

- El proceso de modificación del árbol de transformación especificado en la subcláusula 8.7.2 se invoca con la ubicación de luma (xCb, yCb), la ubicación de luma (xB0, yB0) fijada igual a (0, 0), el tamaño del bloque de codificación de luma nCbS fijado igual a (1 << Log2MaxTamañoCb), la profundidad del árbol de codificación profundidadCu igual a 0, la formación de indicadores de división del árbol de codificación indicador_cu_dividida, la formación de modalidades de partición de la unidad de codificación ModalidadPart, la formación de indicadores de división del árbol de transformación indicador_transformación_dividida, la formación de indicadores de subunidad de predicción indicadorSubPu y el tamaño de la subunidad de predicción tamañoSubPu en las entradas, y la formación modificada de indicadores de división del árbol de transformación indicador_transformación_dividida como salida.

H.8.7.2 Proceso de modificación del árbol de transformación de la unidad de codificación que contiene la subunidad de predicción

[0162] Las entradas para este proceso son:

- una ubicación de luma (xCb, yCb) que especifique la muestra superior izquierda del bloque de codificación de luma actual en relación con la muestra de luma superior izquierda de la imagen actual,
- una ubicación de luma (xB0, yB0) para especificar la muestra superior izquierda del bloque de luma actual en relación con la muestra superior izquierda del bloque de codificación de luma actual,
- tamaño de bloque de codificación de luma nCbS,
- una variable que especifica la profundidad del árbol de codificación profundidadCu,
- formación de indicadores de división del árbol de codificación indicador_cu_dividida,
- formación de indicadores de división del árbol de transformación indicador_transformación_dividida,
- la formación de indicadores de subunidad de predicción indicadorSubPu,
- el tamaño de subunidad de predicción tamañoSubPu,

[0163] La salida de este proceso es la modificada:

- la formación de indicadores de división del árbol de transformación indicador_transformación_dividida,

[0164] Según el valor de indicador_cu_dividida[xCb + xB0] [yCb + yB0] [profundidadCu], vale lo siguiente:

- Si indicador_cu_dividida [xCb + xB0] [yCb + yB0] [profundidadCu] es igual a 1, se aplican las siguientes etapas ordenadas:

  1. Las variables xB1 e yB1 se obtienen de la siguiente manera:

  - La variable xB1 se fija igual a xH0 + (nCbS >> 1).

  - La variable yB1 se fija igual a yB0 + (nCbS >> 1).

  2. El proceso de modificación del árbol de transformación, según lo especificado en esta subcláusula, se invoca con la ubicación de luma (xCb, yCb), la ubicación de luma (xB0, yB0), el tamaño del bloque de codificación de luma nCbS se fija igual a (nCbS >> 1), la profundidad del árbol de codificación profundidadCu se fija igual a profundidadCu + 1, la formación de indicadores de división del árbol de codificación indicador_cu_dividida, la formación de modalidades de partición de la unidad de codificación ModalidadPart, la formación de indicadores de división del árbol de transformación indicador_transformación_dividida, la formación de indicadores de subunidad de predicción indicadorSubPu y el tamaño de la subunidad de predicción tamañoSubPu, como entradas y la formación modificada de indicadores de división del árbol de transformación indicador_transformación_dividida como salida.

  3. El proceso de modificación del árbol de transformación, según lo especificado en esta subcláusula, se invoca con la ubicación de luma (xCb, yCb), la ubicación de luma (xB1, yB0), el tamaño del bloque de codificación de luma nCbS fijado igual a (nCbS >> 1), la profundidad del árbol de codificación profundidadCu fijada igual a profundidadCu + 1, la formación de indicadores de división del árbol de codificación indicador_cu_dividida, la formación de modalidades de partición de la unidad de codificación ModalidadPart, la formación de indicadores de división del árbol de transformación indicador_transformación_dividida, la formación de indicadores de subunidad de predicción indicadorSubPu y el tamaño de la subunidad de predicción tamañoSubPu, como entradas y la formación modificada de indicadores de división del árbol de transformación indicador_transformación_dividida como salida.

  4. El proceso de modificación del árbol de transformación, según lo especificado en esta subcláusula, se invoca con la ubicación de luma (xCb, yCb), la ubicación de luma (xB0, yB1), el tamaño del bloque de codificación de luma nCbS fijado igual a (nCbS >> 1), la profundidad del árbol de codificación profundidadCu fijada igual a profundidadCu + 1, la formación indicadores de división del árbol de codificación indicador_cu_dividida, la formación de modalidades de partición de la unidad de codificación ModalidadPart, la formación de indicadores de división del árbol de transformación indicador_transformación_dividida, la formación de indicadores de subunidad de predicción indicadorSubPu y el tamaño de la subunidad de predicción tamañoSubPu, como entradas y la formación modificada de indicadores de división del árbol de transformación indicador_transformación_dividida como salida.

  5. El proceso de modificación del árbol de transformación, según lo especificado en esta subcláusula, se invoca con la ubicación de luma (xCb, yCb), la ubicación de luma (xB1, yB1), el tamaño del bloque de codificación de luma nCbS fijado igual a (nCbS >> 1), la profundidad del árbol de codificación profundidadCu fijada igual a profundidadCu + 1, la formación de indicadores de división del árbol de codificación indicador_cu_dividida, la formación de modalidades de división de la unidad de codificación ModalidadPart, la formación de indicadores de división de árbol de transformación indicador_transformación_dividida, la formación de indicadores de subunidad de predicción indicadorSubPu y el tamaño de la subunidad de predicción tamañoSubPu, como entradas y la formación modificada de indicadores de división del árbol de transformación indicador_transformación_dividida como salida.

- De lo contrario (indicador_cu_dividida[xCb + xB0][yCb + yB0][profundidadCu] es igual a 0), si nCbS es mayor que tamañoSubPu, se aplican las siguientes etapas ordenadas:

  1. Las variables xB1 e yB1 se obtienen de la siguiente manera:

  - La variable xB1 se fija igual a xB0 + (nCbS >> 1).

  - La variable yB1 se fija igual a yB0 + (nCbS >> 1).

  2. Obtener la variable indicadorDesbloqueoSubPu siguiendo las siguientes etapas ordenadas:

  - El indicadorDesbloqueoSubPu se fija igual a 0.

- Si indicadorSubPu [xCb + xB0] [yCb + yB0] es igual a 1, el indicadorDesbloqueoSubPu se fija igual a 1.
- Si indicadorSubPu [xCb + xB1] [yCb + yB0] es igual a 1, el indicadorDesbloqueoSubPu se fija igual a 1.
- Si indicadorSubPu [xCb + xB0] [yCb + yB1] es igual a 1, el indicadorDesbloqueoSubPu se fija igual a 1.
- Si indicadorSubPu [xCb + xB1] [yCb + yB1] es igual a 1, el indicadorDesbloqueoSubPu se fija igual a 1.
- Si ModalidadPart [xCb + xB0] [yCb + yB0] es igual a PART_nLx2N, o ModalidadPart [xCb + xB0] [yCb + yB0] es igual a PART_nRx2N, o ModalidadPart [xCb + xB0] [yCb + yB0] es igual a PART2XXXXX, o ModalidadPart [xCb + xB0] [yCb + yB0] es igual a PART_2NxnD, el indicadorDesbloqueoSubPu se fija igual a 0.

3. Cuando el indicadorDesbloqueoSubPu es igual a 1, se aplican las siguientes etapas ordenadas:

- si indicador_transformación_dividida[xCb + xB0][yCb + yB0][profundidadCu] es igual a 0, fijar indicador_transformación_dividida[xCb + xB0][yCb + yB0][profundidadCu] para que sea igual a 1.
- El proceso de modificación del árbol de transformación especificado en la subcláusula 8.7.3 se invoca con la ubicación de luma (xCb, yCb), la ubicación de luma (xB0, yB0), la profundidad del árbol de codificación profundidadCu, el tamaño de bloque nCbS, la formación de indicadores de división del árbol de transformación indicador_transformación_dividida, la formación de indicadores de subunidad de predicción de la unidad de predicción indicadorSubPu y la formación de tamaños de subunidad de predicción de la unidad de predicción tamañoSubPu como entradas, y la formación modificada de indicadores de división del árbol de codificación indicador_transformación_dividida como salida.

H.8.7.3 Proceso de modulación del árbol de transformación del bloque de codificación de luma

[0165] Las entradas para este proceso son:

- una ubicación de luma (xCb, yCb) que especifica la muestra superior izquierda del bloque actual de predicción de luma en relación con la muestra de luma superior izquierda de la imagen actual,
- una ubicación de luma (xB0, yB0) que especifica la muestra superior izquierda del bloque actual de luma en relación con la muestra superior izquierda del bloque actual de codificación de luma,
- variables que especifican el tamaño del bloque nCbS,
- una variable que especifica la profundidad del árbol de transformación profundidadTrafo,
- la formación de indicadores de división del árbol de transformación indicador_transformación_dividida,
- la formación de indicadores de subunidad de predicción indicadorSubPu,
- el tamaño de subunidad de predicción tamañoSubPu,

[0166] La salida de este proceso es la modificada:

- formación de indicadores de división del árbol de transformación indicador_transformación_dividida,

[0167] Si nCbS es más grande que tamañoSubPu, se aplican las siguientes etapas ordenadas.

- Las variables xB1 e yB1 se obtienen de la siguiente manera:
  - La variable xB1 se fija igual a xB0 + (nCbS >> 1).
  - La variable yB1 se fija igual a yB0 + (nCbS >> 1).
- Para x en xB0, xB1
  - Para y en yB0, yB1
  - si indicadorSubPu[xCb + x][yCb + y] es igual a 1
  - si indicador_transformación_dividida[xCb + x][yCb + y][profundidadTrafo + 1] es igual a 0

- fijar indicador_transformación_dividida[xCb + x][yCb + y][profundidadTrafo + 1] para que sea igual a 1.

- el proceso de modificación del árbol de transformación especificado en esta subcláusula se invoca con la ubicación de luma (xCb, yCb), la ubicación de luma (xCb + x, yCb + y), la profundidad del árbol de transformación profundidadTrafo se fija igual a profundidadTrafo + 1, el tamaño del bloque nCbS se fija igual a (nCbS >> 1), la formación de indicadores de división del árbol de transformación indicador_transformación_dividida, la formación de indicadores de subunidad de predicción indicadorSubPu y el tamaño de subunidad de predicción tamañoSubPu como entradas, y la formación modificada de indicadores de división del árbol de codificación indicador_transformación_dividida como salida.

- de lo contrario (si indicador_transformación_dividida[xCb + x][yCb + y][profundidadTrafo + 1] es igual a 1),

- el proceso de modificación del árbol de transformación especificado en esta subcláusula se invoca con la ubicación de luma (xCb, yCb), la ubicación de luma (xCb + x, yCb + y), la profundidad del árbol de transformación profundidadTrafo se fija igual a profundidadTrafo + 1, el tamaño del bloque nCbS se fija igual a (nCbS >> 1), la formación de indicadores de división del árbol de transformación indicador_transformación_dividida, la formación de indicadores de subunidad de predicción indicadorSubPu y el tamaño de subunidad de predicción tamañoSubPu como entradas, y la formación modificada de indicadores de división del árbol de codificación indicador_transformación_dividida como salida.

H.8.8 Proceso de filtro en bucle

[0168] La figura 15 es un diagrama de bloques que ilustra un ejemplo del codificador de vídeo 20 que puede configurarse para implementar las técnicas de esta divulgación. El codificador de vídeo 20 puede realizar intracodificación e intercodificación (incluida la codificación entre vistas) de bloques de vídeo dentro de fragmentos de vídeo, por ejemplo, fragmentos tanto de imágenes de textura como de mapas de profundidad. La información de textura generalmente incluye información de luminancia (brillo o intensidad) y crominancia (color, por ejemplo, tonos rojos y tonos azules). En algunos ejemplos, el codificador de vídeo 20 puede determinar las modalidades de codificación en relación con los fragmentos de luminancia, y reutilizar la información de predicción, a partir de la codificación de la información de luminancia para codificar la información de crominancia (por ejemplo, reutilizando la información de partición, las selecciones de modalidad de intrapredicción, los vectores de movimiento o similares). La intracodificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial en el vídeo dentro de una trama o imagen de vídeo dada. La intercodificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal en el vídeo dentro de tramas o imágenes adyacentes de una secuencia de vídeo. La intramodalidad (modalidad I) puede referirse a cualquiera entre varias modalidades de codificación de base espacial. Las intermodalidades, tales como la predicción unidireccional (modalidad P) o la bipredicción (modalidad B), pueden referirse a cualquiera entre varias modalidades de codificación de base temporal.

[0169] Como se muestra en la figura 15, el codificador de vídeo 20 recibe un bloque de vídeo actual (es decir, un bloque de datos de vídeo, tal como un bloque de luminancia, un bloque de crominancia o un bloque de profundidad) dentro de una trama de vídeo (por ejemplo, una imagen de textura o un mapa de profundidad) a codificar. En el ejemplo de la figura 15, el codificador de vídeo 20 incluye una unidad de selección de modalidad 40, una memoria de datos de vídeo 41, un almacén temporal de imágenes decodificadas 64, un sumador 50, una unidad de procesamiento de transformación 52, una unidad de cuantización 54, una unidad de filtro 63 y una unidad de codificación por entropía 56. La unidad de filtro 63 puede aplicar un proceso de filtro de desbloqueo según se describe en esta divulgación. A su vez, la unidad de selección de modalidad 40 incluye una unidad de compensación de movimiento 44, una unidad de estimación de movimiento 42, una unidad de procesamiento de intrapredicción 46 y una unidad de partición 48. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 incluye también una unidad de cuantización inversa 58, una unidad de procesamiento de transformación inversa 60 y un sumador 62. La unidad de filtro 63 puede incluir un filtro de desbloqueo y / o un filtro SAO para filtrar los fronteras del bloque, para eliminar distorsiones de pixelado del vídeo reconstruido. También pueden usarse filtros adicionales (en bucle o pos-bucle), además del filtro de eliminación de bloques. Dichos filtros no se muestran por razones de brevedad pero, si se desea, pueden filtrar la salida del sumador 50 (como un filtro en bucle).

[0170] La memoria de datos de vídeo 41 puede almacenar datos de vídeo a codificar por parte de los componentes del codificador de vídeo 20. Los datos de vídeo almacenados en la memoria de datos de vídeo 41 se pueden obtener, por ejemplo, a partir del origen de vídeo 18. El almacén temporal de imágenes decodificadas 64 puede ser una memoria de imágenes de referencia que almacene datos de vídeo de referencia para su uso en la codificación de datos de vídeo mediante el codificador de vídeo 20, por ejemplo, en modalidades de intracodificación o de intercodificación. La memoria de datos de vídeo 41 y el almacén temporal de imágenes decodificadas 64 pueden estar formados por cualquiera entre varios dispositivos de memoria, tales como memoria dinámica de acceso aleatorio (DRAM), incluyendo DRAM síncrona (SDRAM), RAM magnetorresistiva (MRAM), RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. La memoria de datos de vídeo 41 y el almacén temporal de imágenes decodificadas 64 pueden ser proporcionados por el mismo dispositivo de memoria o dispositivos de memoria independientes. En diversos ejemplos, la memoria de datos de vídeo 41 puede estar en un chip con otros componentes del codificador de vídeo 20, o fuera de chip con respecto a esos componentes.

[0171] Durante el proceso de codificación, el codificador 20 de vídeo recibe una trama o un fragmento de vídeo a codificar. La trama o el fragmento pueden dividirse en múltiples bloques de vídeo. La unidad 42 de estimación de movimiento y la unidad 44 de compensación de movimiento llevan a cabo la codificación de predicción intra del bloque de vídeo recibido con respecto a uno o más bloques en una o más tramas de referencia, para proporcionar la predicción temporal. La unidad de procesamiento de intrapredicción 46, de forma alternativa, puede realizar la codificación intrapredictiva del bloque de vídeo recibido, con respecto a uno o más bloques contiguos en la misma trama o fragmento que el bloque a codificar, para proporcionar predicción espacial. El codificador de vídeo 20 puede llevar a cabo múltiples pases de codificación, por ejemplo, para seleccionar una modalidad de codificación adecuada para cada bloque de datos de vídeo.

[0172] Además, la unidad de división 48 puede dividir bloques de datos de vídeo en subbloques, basándose en la evaluación de los anteriores esquemas de división en las anteriores pases de codificación. Por ejemplo, la unidad de partición 48 puede dividir inicialmente una trama o un fragmento en varias LCU, y dividir cada una de las LCU en varias subCU, basándose en un análisis de velocidad-distorsión (por ejemplo, una optimización de velocidad-distorsión). La unidad de selección de modalidad 40 puede producir además una estructura de datos de árbol cuádruple, indicativa de la división de una LCU en las sub-CU. Las CU de nodos hoja del árbol cuádruple pueden incluir una o más PU y una o más TU.

[0173] La unidad de selección de modalidad 40 puede seleccionar una de las modalidades de codificación, intra o inter, por ejemplo, basándose en los resultados de errores, y proporciona el bloque intracodificado o intercodificado resultante al sumador 50 para generar datos de bloques residuales, y al sumador 62 para reconstruir el bloque codificado para su uso como una trama de referencia. La unidad de selección de modalidad 40 proporciona también elementos sintácticos, tales como vectores de movimiento, indicadores de intramodalidad, información de división y otra información sintáctica de este tipo, a la unidad de codificación por entropía 56.

[0174] La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden estar sumamente integradas, pero se ilustran por separado con fines conceptuales. La estimación del movimiento, realizada por la unidad de estimación de movimiento 42, es el proceso de generación de vectores de movimiento, que estiman el movimiento para los bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de una PU de un bloque de vídeo dentro de una trama o imagen de vídeo actual, con respecto a un bloque predictivo dentro de una trama de referencia (u otra unidad codificada), con respecto al bloque actual que se está codificando dentro de la trama actual (u otra unidad codificada).

[0175] Un bloque predictivo es un bloque que se revela como estrechamente coincidente con el bloque a codificar, en lo que respecta a la diferencia de píxeles, lo cual puede determinarse mediante la suma de diferencias absolutas (SAD), suma de las diferencias al cuadrado (SSD) u otras métricas de diferencia. En algunos ejemplos, el codificador de vídeo 20 puede calcular valores para posiciones de píxeles fraccionarias de imágenes de referencia almacenadas en el almacén temporal de imágenes de referencia 64. Por ejemplo, el codificador de vídeo 20 puede interpolar valores de posiciones de un cuarto de píxel, posiciones de un octavo de píxel u otras posiciones fraccionarias de píxel de la imagen de referencia. Por lo tanto, la unidad de estimación de movimiento 42 puede realizar una búsqueda de movimiento relativa a las posiciones de píxel completas y las posiciones de píxel fraccionarias, y facilitar un vector de movimiento con una precisión de píxel fraccionaria.

[0176] La unidad de estimación de movimiento 42 calcula un vector de movimiento para una PU de un bloque de vídeo en un fragmento intercodificado comparando la posición de la PU con la posición de un bloque predictivo de una imagen de referencia. La imagen de referencia puede seleccionarse a partir de una primera lista de imágenes de referencia (lista 0) o una segunda lista de imágenes de referencia (lista 1), cada una de las cuales identifica una o más imágenes de referencia almacenadas en el almacén temporal de imágenes decodificadas 64. Las listas de imágenes de referencia pueden construirse utilizando las técnicas de esta divulgación. La unidad de estimación de movimiento 42 envía el vector de movimiento calculado a la unidad de codificación por entropía 56 y a la unidad de compensación de movimiento 44.

[0177] La compensación de movimiento, llevada a cabo por la unidad de compensación de movimiento 44, puede implicar capturar o generar el bloque predictivo basándose en el vector de movimiento determinado por la unidad de estimación de movimiento 42. De nuevo, la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden integrarse funcionalmente, en algunos ejemplos. Tras recibir el vector de movimiento para la PU del bloque de vídeo actual, la unidad de compensación de movimiento 44 puede localizar el bloque predictivo al que apunta el vector de movimiento en una de las listas de imágenes de referencia. El sumador 50 forma un bloque de vídeo residual restando los valores de píxel del bloque predictivo a los valores de píxel del bloque de vídeo actual que se está codificando, generando valores de diferencias de píxel, como se expone más adelante. En general, la unidad de estimación de movimiento 42 lleva a cabo la estimación de movimiento con respecto a los componentes de luma, y la unidad de compensación de movimiento 44 utiliza los vectores de movimiento calculados basándose en los componentes de luma, tanto para los componentes de croma como para los componentes de luma. De esta manera, la unidad de compensación de movimiento 44 puede reutilizar la información de movimiento determinada para los componentes de luma para codificar componentes de croma, de modo que la unidad de estimación de movimiento 42

no necesita realizar una búsqueda de movimiento para los componentes de croma. La unidad de selección de modalidad 40 también puede generar elementos sintácticos asociados a los bloques de vídeo y al fragmento de vídeo para su uso por el decodificador de vídeo 30 en la decodificación de los bloques de vídeo del fragmento de vídeo.

[0178] La unidad de procesamiento de intrapredicción 46 puede intrapredecir un bloque actual, como alternativa a la interpredicción realizada por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44, como se ha descrito anteriormente. En particular, la unidad de procesamiento de intrapredicción 46 puede determinar una modalidad de intrapredicción a usar para codificar un bloque actual. En algunos ejemplos, la unidad de procesamiento de intrapredicción 46 puede codificar un bloque actual usando diversas modalidades de intrapredicción, por ejemplo, durante pases de codificación independientes, y la unidad de procesamiento de intrapredicción 46 (o la unidad de selección de modalidad 40, en algunos ejemplos) puede seleccionar una modalidad de intrapredicción adecuada para su uso a partir de las modalidades probadas.

[0179] Por ejemplo, la unidad de procesamiento de intrapredicción 46 puede calcular valores de velocidad-distorsión usando un análisis de velocidad-distorsión para las diversas modalidades de intrapredicción probadas, y seleccionar la modalidad de intrapredicción que tenga las mejores características de velocidad-distorsión entre las modalidades probadas. El análisis de velocidad-distorsión determina, en general, una magnitud de distorsión (o error) entre un bloque codificado y un bloque original no codificado que fue codificado para generar el bloque codificado, así como una velocidad de bits (es decir, un número de bits) usada para generar el bloque codificado. La unidad de procesamiento de intrapredicción 46 puede calcular razones a partir de las distorsiones y velocidades para los diversos bloques codificados, para determinar qué modalidad de intrapredicción presenta el mejor valor de velocidad-distorsión para el bloque.

[0180] Después de seleccionar una modalidad de intrapredicción para un bloque, la unidad de procesamiento de intrapredicción 46 puede proporcionar información, indicativa de la modalidad de intrapredicción seleccionada para el bloque, a la unidad de codificación por entropía 56. La unidad de codificación por entropía 56 puede codificar la información que indica la modalidad de intrapredicción seleccionada. 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 modalidades de intrapredicción y una pluralidad de tablas modificadas de índices de modalidades de intrapredicción (también denominadas tablas de correlación de palabras de código), definiciones de contextos de codificación para diversos bloques e indicaciones de una modalidad de intrapredicción más probable, una tabla de índices de modalidades de intrapredicción y una tabla modificada de índices de modalidades de intrapredicción, a usar para cada uno de los contextos.

[0181] El codificador de vídeo 20 forma un bloque de vídeo residual restando los datos de predicción, de la unidad de selección de modalidad 40, del bloque de vídeo original que se está codificando. El sumador 50 representa el componente o los componentes que realizan esta operación de resta. La unidad de procesamiento de transformación 52 aplica una transformación, tal como una transformación discreta de coseno (DCT) o una transformación conceptualmente similar, al bloque residual, produciendo un bloque de vídeo que comprende valores residuales de coeficientes de transformación. La unidad de procesamiento de transformación 52 puede llevar a cabo otras transformaciones que son conceptualmente similares a la DCT. También se podrían usar transformaciones de ondículas, transformaciones de enteros, transformaciones de subbandas u otros tipos de transformaciones. En cualquier caso, la unidad de procesamiento de transformación 52 aplica la transformación al bloque residual, produciendo un bloque de coeficientes de transformación residuales.

[0182] La transformación puede convertir la información residual desde un dominio de valores de píxel a un dominio de transformación, tal como un dominio de frecuencia. La unidad de procesamiento de transformación 52 puede enviar los coeficientes de transformación resultantes a la unidad de cuantización 54. La unidad de cuantización 54 cuantiza los coeficientes de transformación para reducir más la velocidad de bits. El proceso de cuantización puede reducir la profundidad de bits asociada a algunos, o a la totalidad, de los coeficientes. El grado de cuantización se puede modificar ajustando un parámetro de cuantización. En algunos ejemplos, la unidad de cuantización 54 puede realizar, a continuación, una exploración de la formación que incluye los coeficientes de transformación cuantizados. De forma alternativa, la unidad de codificación por entropía 56 puede realizar la exploración.

[0183] Tras la cuantización, la unidad de codificación por entropía 56 codifica por entropía los coeficientes de transformación cuantizados. Por ejemplo, la unidad de codificación por entropía 56 puede llevar a cabo la codificación de longitud variable adaptativa según el contexto (CAVLC), la codificación aritmética binaria adaptativa según el contexto (CABAC), la codificación aritmética binaria adaptativa según el contexto y basada en sintaxis (SBAC), la codificación por entropía mediante la división en intervalos de probabilidades (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 contiguos. Tras la codificación por entropía por la unidad de codificación por entropía 56, el flujo de bits codificado se puede transmitir a otro dispositivo (por ejemplo, el decodificador de vídeo 30) o archivar para su posterior transmisión o recuperación.

[0184] La unidad de cuantización inversa 58 y la unidad de procesamiento de transformación inversa 60 aplican la cuantización inversa y la transformación inversa, respectivamente, para reconstruir el bloque residual en el dominio

del píxel, por ejemplo, para su posterior uso como bloque de referencia. La unidad de compensación de movimiento 44 puede calcular un bloque de referencia añadiendo el bloque residual a un bloque predictivo de una de las imágenes de referencia almacenadas en el almacén temporal de imágenes decodificadas 64. La unidad de compensación de movimiento 44 también puede aplicar uno o más filtros de interpolación al bloque residual reconstruido para calcular valores de píxel fraccionarios para su uso en la estimación de movimiento. El sumador 62 añade el bloque residual reconstruido al bloque de predicción compensado por movimiento, producido por la unidad de compensación de movimiento 44, para generar un bloque de vídeo reconstruido para su almacenamiento en el almacén temporal de imágenes decodificadas 64. El bloque de vídeo reconstruido puede ser utilizado por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44, como bloque de referencia para intercodificar un bloque en una trama de vídeo posterior.

[0185] El codificador de vídeo 20 puede codificar mapas de profundidad de una manera que se asemeja esencialmente a las técnicas de codificación para codificar componentes de luminancia, si bien sin los correspondientes componentes de crominancia. Por ejemplo, la unidad de procesamiento de intrapredicción 46 puede intrapredecir bloques de mapas de profundidad, mientras que la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden interpredecir bloques de mapas de profundidad. Sin embargo, como se ha expuesto anteriormente, durante la interpredicción de mapas de profundidad, la unidad de compensación de movimiento 44 puede escalar (es decir, ajustar a escala) los valores de los mapas de profundidad de referencia basándose en las diferencias en los rangos de profundidad y los valores de precisión para los rangos de profundidad. Por ejemplo, si diferentes valores de profundidad máxima en el mapa de profundidad actual y un mapa de profundidad de referencia corresponden a la misma profundidad del mundo real, el codificador de vídeo 20 puede ajustar a escala el valor de profundidad máxima del mapa de profundidad de referencia para que sea igual al valor de profundidad máxima en el mapa de profundidad actual, a efectos de predicción. Adicional o alternativamente, el codificador de vídeo 20 puede usar los valores actualizados del rango de profundidad y los valores de precisión para generar una imagen de síntesis de vista para la predicción de síntesis de vista, por ejemplo, utilizando técnicas esencialmente similares a la predicción entre vistas.

[0186] El codificador de vídeo 20 representa un ejemplo de un codificador de vídeo que puede configurarse para realizar cualquiera de las técnicas descritas en esta divulgación, solas o en cualquier combinación. Por ejemplo, el codificador de vídeo 20 puede configurarse para realizar técnicas para la predicción de movimiento a nivel de sub-PU para la codificación de vídeo en la norma 3D-HEVC.

[0187] En un ejemplo, el codificador de vídeo 20 puede configurarse para realizar una modalidad de BVSP de sub-PU para predecir una PU que incluye dos o más sub-PU. En la modalidad de BVSP de sub-PU, la unidad de compensación de movimiento 44 del codificador de vídeo 20 determina la primera información de movimiento para una PU actual que incluye al menos un vector de movimiento de disparidad y un índice de vista de referencia asociado, que identifica una imagen de referencia entre vistas. La unidad de compensación de movimiento 44 luego divide la PU actual en dos o más sub-PU, y determina la segunda información de movimiento para cada una de las sub-PU que incluye al menos un vector de movimiento de disparidad asociado a un bloque de profundidad de la imagen de referencia entre vistas, correspondiente. a cada una de las sub-PU. La unidad de compensación de movimiento 44 realiza la compensación de movimiento para predecir cada una de las sub-PU de la PU en función de la segunda información de movimiento para cada una de las sub-PU. De acuerdo a las técnicas de esta divulgación, para cada PU predicha utilizando la modalidad de BVSP de sub-PU, el codificador de vídeo 20 almacena la segunda información de movimiento obtenida para cada una de las sub-PU de la PU actual, incluso después de que se realiza la compensación de movimiento. La segunda información de movimiento puede almacenarse en el almacén temporal de imágenes decodificadas 64. La información de movimiento adicional, almacenada para la PU actual, se puede usar luego para predecir las PU posteriores, para las cuales la PU actual es un bloque vecino.

[0188] En otro ejemplo, el codificador de vídeo 20 puede configurarse para realizar una modalidad de TMVP avanzada para predecir las sub-PU de una PU en la codificación de capa única, para la cual se puede permitir el refinamiento del vector de movimiento. En la modalidad de TMVP avanzada, la unidad de compensación de movimiento 44 del codificador de vídeo 20 determina un vector de movimiento de primera etapa para una PU actual que identifica un bloque de una imagen de referencia correspondiente a la PU actual. La unidad de compensación de movimiento 44 luego divide la PU actual en dos o más sub-PU, y determina la información de movimiento de la segunda etapa para cada una de las sub-PU a partir del bloque de la imagen de referencia identificada por el vector de movimiento de la primera etapa, donde la información de movimiento de la segunda etapa para cada una de las sub-PU incluye al menos un vector de movimiento y un índice de referencia asociado. La unidad de compensación de movimiento 44 realiza la compensación de movimiento para predecir cada una de las sub-PU por separado, basándose en la información de movimiento de la segunda etapa para cada una de las sub-PU. En algunos ejemplos, la unidad de compensación de movimiento 44 puede determinar una diferencia de vectores de movimiento para refinar el al menos un vector de movimiento de la información de movimiento de la segunda etapa para cada una de las sub-PU.

[0189] En otro ejemplo, el codificador de vídeo 20 puede configurarse para realizar técnicas relacionadas con los procesos de filtro de desbloqueo aplicados a las CU de un bloque de vídeo que incluyen al menos una PU con múltiples sub-PU. De acuerdo a las técnicas de esta divulgación, antes de aplicar un filtro de desbloqueo a una CU del bloque

de vídeo que incluye la PU con las sub-PU, la unidad de filtro 63 del codificador de vídeo 20 convierte la CU para crear fronteras de PU artificiales o fronteras de TU artificiales en las fronteras de sub-PU. En un ejemplo, la unidad de filtro 63 convierte un árbol de transformación de la CU a fin de asociar la PU a una jerarquía de árbol de transformación y asociar cada una de las sub-PU a una TU de modo que las fronteras de sub-PU se conviertan en fronteras de TU artificiales. En otro ejemplo, la unidad de filtro 63 convierte la CU en un árbol de codificación para asociar la PU a una CU y asociar cada una de las sub-PU a una PU de tal manera que los fronteras de la sub-PU se conviertan en fronteras de PU artificiales. La unidad de filtro 63 luego aplica el filtro de desbloqueo a los fronteras de PU entre dos PU adyacentes de la CU y/o las fronteras de TU entre dos TU adyacentes de la CU, incluidas las fronteras de PU artificiales y las fronteras de TU artificiales.

[0190] La figura 16 es un diagrama de bloques que ilustra un ejemplo de decodificador de vídeo 30 que puede implementar las técnicas de esta divulgación. En el ejemplo de la figura 16, el decodificador de vídeo 30 incluye una unidad de decodificación por entropía 70, una memoria de datos de vídeo 71, una unidad de compensación de movimiento 72, una unidad de procesamiento de intrapredicción 74, una unidad de cuantización inversa 76, una unidad de transformación inversa 78, un almacén temporal de imágenes decodificadas 82 y un sumador 80. En algunos ejemplos, el decodificador de vídeo 30 puede realizar un pase de decodificación, en general, recíproco al pase de codificación descrito con respecto al codificador de vídeo 20 (figura 15). La unidad de compensación de movimiento 72 puede generar datos de predicción basándose en vectores de movimiento recibidos desde la unidad de decodificación por entropía 70, mientras que la unidad de procesamiento de intrapredicción 74 puede generar datos de predicción basados en indicadores de modalidad de intrapredicción recibidos desde la unidad de decodificación por entropía 70.

[0191] La memoria de datos de vídeo 71 puede almacenar datos de vídeo, tales como un flujo de bits de vídeo codificado, a decodificar por parte de los componentes del decodificador de vídeo 30. Los datos de vídeo almacenados en la memoria de datos de vídeo 71 pueden obtenerse, por ejemplo, a partir del medio legible por ordenador 16, por ejemplo, desde un origen de vídeo local, tal como una cámara, mediante comunicación de datos de vídeo por red cableada o inalámbrica, o accediendo a medios de almacenamiento físico de datos. La memoria de datos de vídeo 71 puede formar un almacén temporal de imágenes codificadas (CPB) que almacene datos de vídeo codificados a partir de un flujo de bits de vídeo codificado. El almacén temporal de imágenes decodificadas 82 puede ser una memoria de imágenes de referencia que almacene datos de vídeo de referencia para su uso en la decodificación de datos de vídeo mediante el decodificador de vídeo 30, por ejemplo, en modalidades de intracodificación o de intercodificación. La memoria de datos de vídeo 71 y el almacén temporal de imágenes decodificadas 82 pueden estar formados por cualquiera entre varios dispositivos de memoria, tales como memoria dinámica de acceso aleatorio (DRAM), incluyendo DRAM síncrona (SDRAM), RAM magnetorresistiva (MRAM), RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. La memoria de datos de vídeo 71 y el almacén temporal de imágenes decodificadas 82 pueden proporcionarse mediante el mismo dispositivo de memoria o dispositivos de memoria independientes. En diversos ejemplos, la memoria de datos de vídeo 71 puede estar en un chip con otros componentes del decodificador de vídeo 30, o fuera de chip con respecto a esos componentes.

[0192] Durante el proceso de decodificación, el decodificador de vídeo 30 recibe un flujo de bits de vídeo codificado que representa bloques de vídeo de un fragmento de vídeo codificado y elementos sintácticos asociados, desde el codificador de vídeo 20. La unidad de decodificación por entropía 70 del decodificador de vídeo 30 decodifica por entropía el flujo de bits para generar coeficientes cuantizados, vectores de movimiento o indicadores de modalidad de intrapredicción y otros elementos sintácticos. La unidad de decodificación por entropía 70 remite los vectores de movimiento y otros elementos sintácticos a la unidad de compensación de movimiento 72. El decodificador de vídeo 30 puede recibir los elementos sintácticos en el nivel de fragmento de vídeo y/o el nivel de bloque de vídeo.

[0193] Cuando el fragmento de vídeo se codifica como un fragmento intracodificado (I), la unidad de procesamiento de intrapredicción 74 puede generar datos de predicción para un bloque de vídeo del fragmento de vídeo actual, basándose en una modalidad de intrapredicción señalizada y en datos de bloques previamente decodificados de la trama o imagen actual. Cuando la trama de vídeo se codifica como un fragmento intercodificado (es decir, B o P), la unidad de compensación de movimiento 72 genera bloques predictivos para un bloque de vídeo del fragmento de vídeo actual basándose en los vectores de movimiento y en otros elementos sintácticos recibidos desde la unidad de decodificación por entropía 70. Los bloques predictivos se pueden generar a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El decodificador de vídeo 30 puede construir las listas de imágenes de referencia, la Lista 0 y la Lista 1, usando técnicas de esta divulgación, basándose en las imágenes de referencia almacenadas en el almacén temporal de imágenes decodificadas 82. La unidad de compensación de movimiento 72 determina la información de predicción para un bloque de vídeo del fragmento de vídeo actual, analizando sintácticamente los vectores de movimiento y otros elementos sintácticos, y usa la información de predicción para generar los bloques predictivos para el bloque de vídeo actual que se está decodificando. Por ejemplo, la unidad de compensación de movimiento 72 usa algunos de los elementos sintácticos recibidos para determinar una modalidad de predicción (por ejemplo, intrapredicción o interpredicción), usada para codificar los bloques de vídeo del fragmento de vídeo, un tipo de fragmento de interpredicción (por ejemplo, fragmento B o fragmento P), información de construcción para una o más de las listas de imágenes de referencia del fragmento, vectores de movimiento para cada bloque de vídeo intercodificado del fragmento, el estado de interpredicción para cada bloque de vídeo intercodificado del fragmento y otra información, para decodificar los bloques de vídeo en el

fragmento de vídeo actual.

[0194] La unidad de compensación de movimiento 72 también puede realizar la interpolación basándose en filtros de interpolación. La unidad de compensación de movimiento 72 puede usar filtros de interpolación como los usados por el codificador de vídeo 20 durante la codificación de los bloques de vídeo, para calcular valores interpolados para píxeles fraccionarios de bloques de referencia. En este caso, la unidad 72 de compensación de movimiento puede determinar los filtros de interpolación usados por el codificador de vídeo 20 a partir de los elementos sintácticos recibidos y usar los filtros de interpolación para generar bloques predictivos.

[0195] La unidad de cuantización inversa 76 cuantiza inversamente, es decir, decuantiza, los coeficientes de transformación cuantizados proporcionados en el flujo de bits y decodificados por la unidad de decodificación por entropía 70. El proceso de cuantización inversa puede incluir el uso de un parámetro de cuantización QP_Y, calculado por el decodificador de vídeo 30 para cada bloque de vídeo en el fragmento de vídeo, para determinar un grado de cuantización y, asimismo, un grado de cuantización inversa que debería aplicarse. La unidad de procesamiento de transformación inversa 78 aplica una transformación inversa, por ejemplo una DCT inversa, una transformación inversa de enteros o un proceso de transformación inversa conceptualmente similar, a los coeficientes de transformación, con el fin de generar bloques residuales en el dominio del píxel.

[0196] Una vez que la unidad de compensación de movimiento 72 ha generado el bloque predictivo para el bloque de vídeo actual basándose en los vectores de movimiento y en otros elementos sintácticos, el decodificador de vídeo 30 forma un bloque de vídeo decodificado sumando los bloques residuales procedentes de la unidad de procesamiento de transformación inversa 78 a los correspondientes bloques predictivos generados por la unidad de compensación de movimiento 72. El sumador 90 representa el componente o los componentes que realizan esta operación de suma. La unidad de filtro 63 puede aplicar un proceso de filtro de desbloqueo. La unidad de filtro 63 puede incluir un filtro de desbloqueo y / o un filtro SAO para filtrar los fronteras del bloque, para eliminar distorsiones de pixelado del vídeo reconstruido. También pueden usarse filtros adicionales (en bucle o pos-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 80 (como un filtro en bucle). Los bloques de vídeo decodificados en una trama o imagen determinada se almacenan a continuación en el almacén temporal de imágenes decodificadas 82, que almacena imágenes de referencia usadas para una posterior compensación de movimiento. El almacén temporal de imágenes decodificadas 82 almacena también vídeo decodificado para su presentación posterior en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la figura 1.

[0197] El decodificador de vídeo 30 puede decodificar mapas de profundidad de una manera que se asemeja esencialmente a las técnicas de decodificación para decodificar componentes de luminancia, aunque sin los componentes de crominancia correspondientes. Por ejemplo, la unidad de procesamiento de intrapredicción 74 puede intrapredecir bloques de mapas de profundidad, mientras que la unidad de compensación de movimiento 72 puede interpredecir bloques de mapas de profundidad. Sin embargo, como se ha expuesto anteriormente, durante la interpredicción de mapas de profundidad, la unidad de compensación de movimiento 72 puede escalar (es decir, ajustar a escala) los valores de los mapas de profundidad de referencia basándose en las diferencias en los rangos de profundidad y los valores de precisión para los rangos de profundidad. Por ejemplo, si diferentes valores de profundidad máxima en el mapa de profundidad actual y un mapa de profundidad de referencia corresponden a la misma profundidad del mundo real, el decodificador de vídeo 30 puede ajustar a escala el valor de profundidad máxima del mapa de profundidad de referencia para que sea igual al valor de profundidad máxima en el mapa de profundidad actual, a efectos de predicción. Adicional o alternativamente, el decodificador de vídeo 30 puede usar los valores actualizados del rango de profundidad y los valores de precisión para generar una imagen de síntesis de vista para la predicción de síntesis de vista, por ejemplo, utilizando técnicas esencialmente similares a la predicción entre vistas.

[0198] El codificador de vídeo 30 representa un ejemplo de un decodificador de vídeo que puede configurarse para realizar cualquiera de las técnicas descritas en esta divulgación, solas o en cualquier combinación. Por ejemplo, el decodificador de vídeo 30 puede configurarse para realizar técnicas para la predicción de movimiento a nivel de sub-PU para la codificación de vídeo en 3D-HEVC.

[0199] En un ejemplo, el decodificador de vídeo 30 se puede configurar para realizar una modalidad BVSP de sub-PU para predecir una PU que incluya dos o más sub-PU. En la modalidad BVSP de sub-PU, la unidad de compensación de movimiento 72 del decodificador de vídeo 30 determina la primera información de movimiento para una PU actual que incluye al menos un vector de movimiento de disparidad y un índice de vista de referencia asociado que identifica una imagen de referencia entre vistas. La unidad de compensación de movimiento 72 luego divide la PU actual en dos o más sub-PU, y determina la segunda información de movimiento para cada una de las sub-PU, que incluye al menos un vector de movimiento de disparidad asociado a un bloque de profundidad de la imagen de referencia entre vistas correspondiente a cada una de las sub-PU. La unidad de compensación de movimiento 72 realiza la compensación de movimiento para predecir cada una de las sub-PU de la PU basándose en la segunda información de movimiento para cada una de las sub-PU. De acuerdo a las técnicas de esta divulgación, para cada PU predicha utilizando la modalidad de BVSP de sub-PU, el decodificador de vídeo 30 almacena la segunda información de movimiento obtenida para cada una de las sub-PU de la PU actual, incluso después de que se realiza la compensación de movimiento. La segunda información de movimiento puede almacenarse en el almacén temporal

de imágenes decodificadas 82. La información de movimiento adicional, almacenada para la PU actual, se puede usar luego para predecir las PU posteriores, para las cuales la PU actual es un bloque vecino.

[0200] En otro ejemplo, el decodificador de vídeo 30 puede configurarse para realizar una modalidad de TMVP avanzada para predecir las sub-PU de una PU en una codificación de capa única, para la cual se puede permitir el refinamiento del vector de movimiento. En la modalidad de TMVP avanzada, la unidad de compensación de movimiento 72 del decodificador de vídeo 30 determina un vector de movimiento de primera etapa para una PU actual que identifica un bloque de una imagen de referencia correspondiente a la PU actual. La unidad de compensación de movimiento 72 luego divide la PU actual en dos o más sub-PU, y determina la información de movimiento de la segunda etapa para cada una de las sub-PU a partir del bloque de la imagen de referencia identificada por el vector de movimiento de la primera etapa, donde la información de movimiento de la segunda etapa para cada una de las sub-PU incluye al menos un vector de movimiento y un índice de referencia asociado. La unidad de compensación de movimiento 72 realiza la compensación de movimiento para predecir cada una de las sub-PU por separado, en función de la información de movimiento de la segunda etapa para cada una de las sub-PU. En algunos ejemplos, la unidad de compensación de movimiento 72 puede determinar una diferencia de vectores de movimiento para refinar el al menos un vector de movimiento de la información de movimiento de la segunda etapa para cada una de las sub-PU.

[0201] En otro ejemplo, el decodificador de vídeo 30 puede configurarse para realizar técnicas relacionadas con los procesos de filtro de desbloqueo aplicados a las CU de un bloque de vídeo que incluyen al menos una PU con múltiples sub-PU. De acuerdo a las técnicas de esta divulgación, antes de aplicar un filtro de desbloqueo a una CU del bloque de vídeo que incluye la PU con las sub-PU, la unidad de filtro 83 del decodificador de vídeo 30 convierte la CU para crear fronteras de PU artificiales o fronteras de TU artificiales en los fronteras de sub-PU. En un ejemplo, la unidad de filtro 83 convierte un árbol de transformación de la CU a fin de asociar la PU a una jerarquía de árbol de transformación y asociar cada una de las sub-PU a una TU de modo que las fronteras de sub-PU se conviertan en fronteras de TU artificiales. En otro ejemplo, la unidad de filtro 83 convierte la CU en un árbol de codificación a fin de asociar la PU a una CU y asociar cada una de las sub-PU a una PU de tal manera que las fronteras de sub-PU se conviertan en fronteras de PU artificiales. La unidad de filtro 83 luego aplica el filtro de desbloqueo a las fronteras de PU entre dos PU adyacentes de la CU y/o las fronteras de TU entre dos TU adyacentes de la CU, incluidas las fronteras de PU artificiales y las fronteras de TU artificiales.

[0202] La figura 17 es un diagrama de flujo que ilustra una operación ejemplar de predicción de una PU actual utilizando una modalidad de BVSP de sub-PU, y de almacenamiento de la información de movimiento de sub-PU determinada. La operación ilustrada se describe en esta divulgación con respecto al decodificador de vídeo 30 de la figura 16. En otros ejemplos, la operación ilustrada puede ser realizada por el codificador de vídeo 20 de la figura 15, o cualquier otro dispositivo de codificación o decodificación que funcione de acuerdo a la norma 3D-HEVC.

[0203] El decodificador de vídeo 30 recibe un flujo de bits de vídeo codificado que representa bloques de vídeo de un fragmento de vídeo codificado y elementos sintácticos asociados. La unidad de decodificación por entropía 70 del decodificador de vídeo 30 decodifica el flujo de bits para generar coeficientes de transformación cuantizados, información de movimiento e indicadores de modalidad de predicción y otros elementos sintácticos. La unidad de decodificación por entropía 70 envía el coeficiente de transformación cuantizado decodificado a la unidad de cuantización inversa 76 y a la unidad de procesamiento de transformación inversa 78 para reconstruir los bloques residuales de los bloques de vídeo a decodificar. La unidad de decodificación 70 envía la información de movimiento decodificada y los indicadores de modalidad de interpredicción a la unidad de compensación de movimiento 72.

[0204] La unidad de compensación de movimiento 72 predice cada PU de cada CU de los bloques de vídeo a decodifica, de acuerdo a la modalidad indicada, entre las modalidades de interpredicción de fusión o de AMVP. Por ejemplo, en la modalidad de fusión, la unidad de compensación de movimiento 72 puede generar una lista de candidatas, en modalidad de fusión, de información de movimiento, que incluye información de movimiento, es decir, vectores de movimiento e índices de referencia asociados, de bloques vecinos espaciales y temporales de la PU actual. En este caso, la información de movimiento decodificada puede incluir un índice de fusión que indica uno de los conjuntos de información de movimiento en la lista de candidatas en la modalidad de fusión, para predecir la PU actual. En la modalidad de BVSP, la lista de candidatos de la modalidad de fusión incluye un candidato especial de BVSP que tiene información de movimiento que incluye un vector de movimiento de disparidad y un índice de vista de referencia asociado, y la información de profundidad se usa para refinar la información de movimiento.

[0205] De acuerdo a las técnicas de esta divulgación, la unidad de compensación de movimiento 72 determina la primera información de movimiento para una PU actual a partir de bloques vecinos de la PU según la modalidad de BVSP en la que la primera información de movimiento incluye al menos un vector de movimiento de disparidad y un índice de vista de referencia asociado que identifica una imagen de referencia entre vistas (140). Como se ha descrito anteriormente, la unidad de compensación de movimiento 72 puede seleccionar la primera información de movimiento para la PU actual como candidata de BVSP a partir de la lista de candidatos de modalidad de fusión. En algunos casos, la primera información de movimiento para la PU actual puede incluir un vector de movimiento de disparidad correspondiente a cada una de las listas primera y segunda de imágenes de referencia.

[0206] La unidad de compensación de movimiento 72 luego divide la PU actual en dos o más sub-PU (142). La unidad de compensación de movimiento 72 determina la segunda información de movimiento para cada una de las sub-PU en las que la segunda información de movimiento incluye al menos un vector de movimiento de disparidad asociado a un bloque de profundidad de la imagen de referencia entre vistas, correspondiente a cada una de las sub-PU (144). Por ejemplo, la unidad de compensación de movimiento 72 puede seleccionar un valor máximo de los cuatro píxeles de esquina para el bloque de profundidad de la imagen de referencia entre vistas, correspondiente a cada una de las sub-PU, y convertir el valor máximo en un componente horizontal del vector de movimiento de disparidad para cada una de las sub-PU. El componente vertical del vector de movimiento de disparidad para cada una de las sub-PU es igual a cero. En algunos casos, la segunda información de movimiento para cada una de las sub-PU puede incluir un vector de movimiento de disparidad correspondiente a cada una de las listas primera y segunda de imágenes de referencia.

[0207] La unidad de compensación de movimiento 72 realiza la compensación de movimiento para predecir cada una de las sub-PU de la PU actual a partir de la imagen de referencia entre vistas, basándose en la segunda información de movimiento (146). Después de realizar la compensación de movimiento, el decodificador de vídeo 30 almacena la segunda información de movimiento para cada una de las sub-PU de la PU actual en una memoria, por ejemplo, el almacén temporal de imágenes decodificadas 82 del decodificador de vídeo 30, para ser usada para predecir las PU posteriores (148). Por ejemplo, el decodificador de vídeo 30 puede almacenar el vector de movimiento de disparidad, obtenido para cada una de las sub-PU en el almacén temporal de imágenes decodificadas 82, asociado a una lista de imágenes de referencia que incluye la imagen de referencia entre vistas, identificada por el índice de vista de referencia de la primera información de movimiento para la PU. Después de generar un bloque predictivo para cada una de las sub-PU de la PU durante la compensación de movimiento, el decodificador de vídeo 30 genera una versión reconstruida del bloque de vídeo basándose en una versión reconstruida de un bloque residual correspondiente y el bloque predictivo para cada una de las sub-PU.

[0208] Convencionalmente, en la modalidad de BVSP de la norma 3D-HEVC, después de realizar la compensación de movimiento para predecir cada una de las sub-PU, solo se almacena un único vector de movimiento de disparidad, correspondiente a cada lista de imágenes de referencia, para la PU actual. El único vector de movimiento de disparidad se almacena para cada lista de imágenes de referencia, incluso aunque la compensación de movimiento de la PU se basa en múltiples vectores de movimiento para las múltiples sub-PU de la PU. En este caso, cuando se usa la PU actual para predecir una PU posterior, se considera que las sub-PU de la PU actual tienen los mismos vectores de movimiento, de modo que la modalidad de BVSP de sub-PU tenga poco impacto en la mejora de la precisión de la predicción del vector de movimiento.

[0209] De acuerdo a las técnicas de esta divulgación, para cada PU codificada en la modalidad de BVSP de sub-PU, el decodificador de vídeo 30 almacena la segunda información de movimiento obtenida para cada una de las sub-PU de la PU actual, incluso después de que se realiza la compensación de movimiento. La información de movimiento adicional, almacenada para la PU actual, se puede usar luego para predecir las PU posteriores, para las cuales la PU actual es un bloque vecino. Por ejemplo, la unidad de compensación de movimiento 72 puede generar una lista de candidatos, en modalidad de fusión, de información de movimiento para predecir una PU posterior que incluye la segunda información de movimiento para al menos una de las sub-PU de la PU almacenada en el almacén temporal de imágenes decodificadas 82 como una candidata de BVSP de sub-PU en la lista de candidatas en modalidad de fusión. En un ejemplo, si la siguiente PU se codifica en la modalidad de BVSP de sub-PU, la unidad de compensación de movimiento 72 puede no necesitar obtener la segunda información de movimiento para cada una de las sub-PU de la PU posterior. En cambio, la unidad de compensación de movimiento 72 puede, en cambio, seleccionar la segunda información de movimiento como candidata de BVSP de sub-PU de la lista de candidatas, en modalidad de fusión, de información de movimiento para predecir las sub-PU de la PU subsiguiente.

[0210] La figura 18 es un diagrama de flujo que ilustra una operación ejemplar de aplicación de un filtro de desbloqueo a cada CU de un bloque de vídeo para filtrar las fronteras de TU y las fronteras de PU, incluidas las fronteras de sub-PU dentro de la CU. El filtro de desbloqueo se aplica después de reconstruir el bloque de vídeo y antes de almacenar el bloque de vídeo en un almacén temporal de imágenes decodificadas como un bloque de una imagen de referencia. La operación ilustrada se describe en esta divulgación con respecto al decodificador de vídeo 30 de la figura 16. En otros ejemplos, la operación ilustrada se puede realizar en el bucle de reconstrucción de bloques de vídeo del codificador de vídeo 20 de la figura 15, o cualquier otro dispositivo de codificación o decodificación que utilice un diseño de sub-PU y filtros de desbloqueo.

[0211] El decodificador de vídeo 30 genera una versión reconstruida de un bloque de vídeo basándose en una versión reconstruida de un bloque residual correspondiente y un bloque predictivo (150). El bloque de vídeo incluye al menos una CU, y la CU puede incluir al menos una PU que se divide en dos o más sub-PU. Como se ha descrito anteriormente con respecto a la figura 17, la unidad de compensación de movimiento 72 del decodificador de vídeo 30 puede generar un bloque predictivo durante la compresión de vídeo para cada una de las sub-PU de la PU. Los filtros de desbloqueo para la norma HEVC no están diseñados para filtrar dentro de una PU, es decir, a lo largo de los fronteras de sub-PU, porque para los bloques de la HEVC se supone que la compensación de movimiento es la misma para toda la PU. Esta divulgación describe técnicas para convertir una PU con sub-PU en una estructura favorecedora del desbloqueo, de modo que los filtros de desbloqueo de la HEVC puedan seguir utilizándose

para los bloques de la 3D-HEVC.

[0212] Antes de aplicar un filtro de desbloqueo a la CU del bloque de vídeo que incluye la PU con las dos o más sub-PU, la unidad de filtro 83 del decodificador de vídeo 30 convierte la CU para crear fronteras de PU artificiales o fronteras de TU artificiales en las fronteras de sub-PU entre dos sub-PU adyacentes de la PU (152). En un ejemplo, la unidad de filtro 83 convierte un árbol de transformación de la CU a fin de asociar la PU a una jerarquía de árbol de transformación y asociar cada una de las sub-PU a una TU de modo que las fronteras de sub-PU se conviertan en fronteras de TU artificiales. En otro ejemplo, la unidad de filtro 83 convierte la CU en un árbol de codificación a fin de asociar la PU a una CU y asociar cada una de las sub-PU a una PU de tal manera que las fronteras de sub-PU se conviertan en fronteras de PU artificiales.

[0213] La unidad de filtro 83 luego aplica el filtro de desbloqueo a las fronteras de PU entre dos PU adyacentes de la CU y/o las fronteras de TU entre dos TU adyacentes de la CU, incluidas los fronteras de PU artificiales y las fronteras de TU artificiales (154). Después de filtrar cada una de las CU de la versión reconstruida del bloque de vídeo, la unidad de filtro 83 almacena el bloque de vídeo en el almacén temporal de imágenes decodificadas 82 como un bloque de una imagen de referencia (156).

[0214] La figura 19 es un diagrama de flujo que ilustra una operación ejemplar de una modalidad de TMVP avanzada para predecir las sub-PU de una PU en codificación de capa única. La operación ilustrada se describe en esta divulgación con respecto al decodificador de vídeo 30 de la figura 16. En otros ejemplos, la operación ilustrada puede realizarse mediante el codificador de vídeo 20 de la Figura 15, o cualquier otro dispositivo de codificación o decodificación que use un diseño de sub-PU.

[0215] El decodificador de vídeo 30 recibe un flujo de bits de vídeo codificado que representa bloques de vídeo de un fragmento de vídeo codificado y elementos sintácticos asociados. La unidad de decodificación por entropía 70 del decodificador de vídeo 30 decodifica el flujo de bits para generar coeficientes de transformación cuantizados, información de movimiento e indicadores de modalidad de predicción y otros elementos sintácticos. La unidad de decodificación por entropía 70 envía el coeficiente de transformación cuantizado decodificado a la unidad de cuantización inversa 76 y a la unidad de procesamiento de transformación inversa 78 para reconstruir los bloques residuales de los bloques de vídeo a decodificar. La unidad de decodificación 70 envía la información de movimiento decodificada y los indicadores de modalidad de interpredicción a la unidad de compensación de movimiento 72.

[0216] La unidad de compensación de movimiento 72 predice cada PU de cada CU de los bloques de vídeo a decodifica, de acuerdo a la modalidad indicada, entre las modalidades de interpredicción de fusión o de AMVP. Por ejemplo, en la modalidad de fusión, la unidad de compensación de movimiento 72 puede generar una lista de candidatas, en modalidad de fusión, de información de movimiento, que incluye información de movimiento, es decir, vectores de movimiento e índices de referencia asociados, de bloques vecinos espaciales y temporales de la PU actual. En este caso, la información de movimiento decodificada puede incluir un índice de fusión que indica uno de los conjuntos de información de movimiento en la lista de candidatas en la modalidad de fusión, para predecir la PU actual. En otro ejemplo, en la modalidad de AMVP, la unidad de compensación de movimiento 72 puede generar una lista de candidatos de la modalidad de AMVP que solo incluye vectores de movimiento de los bloques vecinos espaciales y temporales de la PU actual. En este caso, la información de movimiento decodificada puede incluir un índice de AMVP que indica uno de los vectores de movimiento en la lista de candidatos de la modalidad de AMVP, y también incluir un índice de referencia explícitamente señalizado y cualquier diferencia de vectores de movimiento para refinar el vector de movimiento seleccionado para predecir la PU actual.

[0217] Convencionalmente, el diseño de sub-PU solo está habilitado para la predicción entre capas o entre vistas utilizando la modalidad de interpredicción de fusión. Esta divulgación describe una modalidad de TMVP avanzada para predecir las sub-PU de una PU en la codificación de una sola capa para la cual se puede admitir el refinamiento del vector de movimiento. En un ejemplo, para indicar el rendimiento de la modalidad de TMVP avanzada para predecir la PU actual, la unidad de compensación de movimiento 72 puede generar un candidato de TMVP avanzado en la lista de candidatos en la modalidad de fusión para la PU, donde la selección del candidato de TMVP avanzado indica el rendimiento de la modalidad de TMVP avanzado para predecir la PU.

[0218] De acuerdo a las técnicas de esta divulgación, la unidad de compensación de movimiento 72 determina un vector de movimiento de primera etapa para una PU actual a partir de bloques vecinos de la PU, que identifica un bloque de una imagen de referencia correspondiente a la PU actual (160). Como ejemplo, la unidad de compensación de movimiento 72 puede obtener el vector de movimiento de la primera etapa de los bloques adyacentes espaciales y/o los bloques vecinos temporales de la PU actual. Como otro ejemplo, la unidad de compensación de movimiento 72 puede seleccionar el vector de movimiento de la primera etapa a partir de una lista de candidatos en modalidad de fusión para la PU actual. En otros ejemplos, el vector de movimiento de la primera etapa se puede configurar para que sea un valor constante o predefinido.

[0219] La unidad de compensación de movimiento 72 luego divide la PU actual en dos o más sub-PU (162). La unidad de compensación de movimiento 72 determina la información de movimiento de la segunda etapa para cada

una de las sub-PU a partir del bloque de la imagen de referencia identificada por el vector de movimiento de la primera etapa, donde la información de movimiento de la segunda etapa, para cada una de las sub-PU, incluye al menos un vector de movimiento y un índice de referencia asociado (164). En algunos casos, la información de movimiento de la segunda etapa para cada una de las sub-PU puede incluir un vector de movimiento correspondiente a cada una de las listas primera y segunda de imágenes de referencia. En el caso de que al menos uno de los vectores de movimiento de la información de movimiento de la segunda etapa para una de las sub-PU no esté disponible, la unidad de compensación de movimiento 72 puede usar un vector de movimiento representativo, tal como el vector de movimiento de la primera etapa, para el vector de las sub-PU.

[0220] En algunos ejemplos, en lugar de operar únicamente en la modalidad de fusión, se puede habilitar el refinamiento del vector de movimiento, similar a la modalidad de AMVP. Por ejemplo, la unidad de compensación de movimiento 72 puede determinar los datos de refinamiento de movimiento para la PU actual predicha utilizando la modalidad de TMVP avanzada. En un caso, la unidad de compensación de movimiento 72 puede determinar una diferencia de vectores de movimiento aplicable a la información de movimiento de la segunda etapa para una o más de las sub-PU. Como se ha descrito anteriormente, la diferencia de vectores de movimiento se puede señalizar en la información de movimiento incluida en el flujo de bits de vídeo codificado. En este ejemplo, para indicar el rendimiento de la modalidad de TMVP avanzada utilizando los datos de refinamiento de movimiento para predecir la PU actual, la unidad de compensación de movimiento 72 puede generar un candidato de TMVP avanzada en la lista de candidatos en modalidad de AMVP para la PU, donde la selección del candidato de TMVP avanzada indica el rendimiento de la modalidad de TMVP avanzada utilizando los datos de refinamiento de movimiento para predecir la PU.

[0221] La unidad de compensación de movimiento 72 realiza la compensación de movimiento para predecir cada una de las sub-PU por separado, basándose en la información de movimiento de la segunda etapa para cada una de las sub-PU (166). Después de generar un bloque predictivo para cada una de las sub-PU de la PU durante la compensación de movimiento, el decodificador de vídeo 30 genera una versión reconstruida del bloque de vídeo basándose en una versión reconstruida de un bloque residual correspondiente y el bloque predictivo para cada una de las sub-PU.

[0222] En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, firmware o cualquier combinación de los mismos. Si se implementan en software, las funciones se pueden almacenar en, o transmitir por, un medio legible por ordenador, como una o más instrucciones o código, y ejecutarse mediante una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que correspondan a un medio tangible tal como medios de almacenamiento de datos o medios de comunicación que incluyan cualquier medio que facilite la transferencia de un programa informático desde un lugar a otro, por ejemplo, de acuerdo a un protocolo de comunicación. De esta manera, los medios legibles por ordenador pueden corresponder en general a (1) medios de almacenamiento tangibles legibles por ordenador que sean no transitorios o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser medios disponibles cualesquiera a los que se pueda acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en esta divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

[0223] A modo de ejemplo, y no de manera limitativa, tales medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento de disco óptico, almacenamiento de disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que pueda usarse para almacenar código de programa deseado en forma de instrucciones o estructuras de datos y al que pueda accederse mediante un ordenador. Además, cualquier conexión recibe debidamente la denominación de medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde una sede de la Red, un servidor u otro origen remoto usando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas se incluyen en la definición de medio. Sin embargo, debería entenderse que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que, en cambio, se orientan a medios de almacenamiento tangibles no transitorios. El término disco, como se usa en el presente documento, incluye un disco compacto (CD), un disco láser, un disco óptico, un disco versátil digital (DVD), un disco flexible y un disco Blu-ray, donde algunos discos normalmente emiten datos magnéticamente, mientras que otros discos emiten datos ópticamente con láseres. Las combinaciones de los anteriores también deberían incluirse dentro del alcance de los medios legibles por ordenador.

[0224] Las instrucciones pueden ser ejecutadas por uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), formaciones de compuertas programables in situ (FPGA) u otros circuitos lógicos, integrados o discretos, equivalentes. En consecuencia, el término "procesador", como se usa en el presente documento, se puede referir a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento se puede proporcionar dentro de módulos de hardware y/o software dedicados, configurados para la codificación y la

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

[0225] Las técnicas de la presente divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluidos un equipo manual inalámbrico, un circuito integrado (CI) o un conjunto de CI (por ejemplo, un conjunto de chips). Diversos componentes, módulos o unidades se describen en esta divulgación para enfatizar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no requieren necesariamente su realización mediante diferentes unidades de hardware. En cambio, como se ha descrito anteriormente, diversas unidades se pueden combinar en una unidad de hardware de códec, o ser proporcionadas por un grupo de unidades de hardware interoperativas, incluyendo uno o más procesadores, como se ha descrito anteriormente, conjuntamente con software y/o firmware adecuados.

[0226] Se han descrito diversos ejemplos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.

REIVINDICACIONES

1. Un procedimiento de procesamiento de datos de vídeo, comprendiendo el procedimiento:

determinar primera información de movimiento para una unidad de predicción (PU) (173) de una unidad de codificación (CU) de un bloque de vídeo a partir de bloques vecinos de la PU de acuerdo con una modalidad de predicción de síntesis de retrovisión (BVSP), en el que determinar la primera información de movimiento para la PU comprende seleccionar la primera información de movimiento como un candidato de BVSP a partir de una lista de candidatos en modalidad de fusión de información de movimiento para predecir la PU, y en el que la primera información de movimiento incluye al menos un vector de movimiento de disparidad (187) y un índice de vista de referencia asociado que identifica una imagen de referencia entre vistas (182);

dividir la PU en dos o más sub-PU (185);

determinar segunda información de movimiento para cada una de las sub-PU, en el que la segunda información de movimiento incluye al menos un vector de movimiento de disparidad (186) asociado a un bloque de profundidad (174) de la imagen de referencia entre vistas correspondiente a cada una de las sub-PU;

realizar una compensación de movimiento para predecir cada una de las sub-PU a partir de la imagen de referencia entre vistas basándose en la segunda información de movimiento;

almacenar la segunda información de movimiento para cada una de las sub-PU de la PU en una memoria, que se va a usar para predecir PU posteriores; y

predecir una PU posterior basándose en la segunda información de movimiento almacenada para al menos una de las sub-PU de la PU, en el que la PU posterior es un bloque vecino de la PU, y en el que predecir la PU posterior comprende generar una lista de candidatos en modalidad de fusión de información de movimiento para predecir la PU posterior, donde la lista de candidatos en modalidad de fusión incluye la segunda información de movimiento para al menos una de las sub-PU de la PU almacenada en la memoria como un candidato de BVSP de sub-PU en la lista de candidatos en modalidad de fusión.

2. El procedimiento de la reivindicación 1, en el que determinar la segunda información de movimiento para cada una de las sub-PU comprende obtener el al menos un vector de movimiento de disparidad para que cada una de las sub-PU tenga un componente horizontal igual a un valor de profundidad máximo de cuatro píxeles de esquina para el bloque de profundidad de la imagen de referencia entre vistas correspondiente a cada una de las sub-PU y un componente vertical igual a cero.

3. El procedimiento de la reivindicación 1, en el que determinar la segunda información de movimiento para cada una de las sub-PU comprende seleccionar la segunda información de movimiento como un candidato de BVSP de sub-PU a partir de una lista de candidatos en modalidad de fusión de información de movimiento para predecir las sub-PU.

4. El procedimiento de la reivindicación 1, en el que la memoria comprende un almacén temporal de imágenes decodificadas, y en el que almacenar la segunda información de movimiento comprende almacenar el vector de movimiento de disparidad para cada una de las sub-PU en el almacén temporal de imágenes decodificadas con una lista de imágenes de referencia que incluye la imagen de referencia entre vistas identificada por el índice de vista de referencia de la primera información de movimiento para la PU.

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

generar un bloque predictivo para cada una de las sub-PU de la PU a partir de la imagen de referencia entre vistas basándose en la segunda información de movimiento;

generar un bloque residual basándose en el bloque de vídeo y el bloque predictivo para cada una de las sub-PU; y

codificar el bloque residual y un indicador de al menos la primera información de movimiento para la PU en un flujo de bits de vídeo.

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

decodificar un bloque residual y un indicador de al menos la primera información de movimiento para la PU a partir de un flujo de bits de vídeo recibido;

generar un bloque predictivo para cada una de las sub-PU de la PU a partir de la imagen de referencia entre

vistas basándose en la segunda información de movimiento; y

generar una versión reconstruida del bloque de vídeo basándose en el bloque residual y el bloque predictivo para cada una de las sub-PU.

7. Un dispositivo de procesamiento de vídeo, que comprende:

una memoria configurada para almacenar datos de vídeo; y

uno o más procesadores en comunicación con la memoria, y configurados para:

determinar primera información de movimiento para una unidad de predicción (PU) (173) de una unidad de codificación (CU) de un bloque de vídeo a partir de bloques vecinos de la PU de acuerdo con una modalidad de predicción de síntesis de retrovisión (BVSP), en el que determinar la primera información de movimiento para la PU comprende seleccionar la primera información de movimiento como un candidato de BVSP a partir de una lista de candidatos en modalidad de fusión de información de movimiento para predecir la PU, y en el que la primera información de movimiento incluye al menos un vector de movimiento de disparidad (187) y un índice de vista de referencia asociado que identifica una imagen de referencia entre vistas (182),

dividir la PU en dos o más sub-PU (185),

determinar segunda información de movimiento para cada una de las sub-PU, em el que la segunda información de movimiento incluye al menos un vector de movimiento de disparidad (186) asociado a un bloque de profundidad (174) de la imagen de referencia entre vistas, correspondiente a cada una de las sub-PU,

realizar una compensación de movimiento para predecir cada una de las sub-PU a partir de la imagen de referencia entre vistas basándose en la segunda información de movimiento,

donde la memoria está configurada para almacenar la segunda información de movimiento para cada una de las sub-PU de la PU en la memoria, que se va a usar para predecir PU posteriores; y

predecir una PU posterior basándose en la segunda información de movimiento almacenada para al menos una de las sub-PU de la PU, en el que la PU posterior es un bloque vecino de la PU, y en el que el uno o más procesadores están configurados para generar una lista de candidatos en modalidad de fusión de información de movimiento para predecir la PU posterior, donde la lista de candidatos en modalidad de fusión incluye la segunda información de movimiento para al menos una de las sub-PU de la PU almacenada en la memoria como un candidato de BVSP de sub-PU en la lista de candidatos en modalidad de fusión.

8. El dispositivo de la reivindicación 7, en el que el dispositivo de procesamiento de vídeo comprende un dispositivo de codificación de vídeo, y en el que el uno o más procesadores están configurados para:

generar un bloque predictivo para cada una de las sub-PU de la PU a partir de la imagen de referencia entre vistas basándose en la segunda información de movimiento;

generar un bloque residual basándose en el bloque de vídeo y el bloque predictivo para cada una de las sub-PU; y

codificar el bloque residual y un indicador de al menos la primera información de movimiento para la PU en un flujo de bits de vídeo.

9. El dispositivo de la reivindicación 7, en el que el dispositivo de procesamiento de vídeo comprende un dispositivo de decodificación de vídeo, y en el que el uno o más procesadores están configurados para:

decodificar un bloque residual y un indicador de al menos la primera información de movimiento para la PU a partir de un flujo de bits de vídeo recibido;

generar un bloque predictivo para cada una de las sub-PU de la PU a partir de la imagen de referencia entre vistas basándose en la segunda información de movimiento; y

generar una versión reconstruida del bloque de vídeo basándose en el bloque residual y el bloque predictivo para cada una de las sub-PU.

10. Un medio de almacenamiento legible por ordenador que almacena instrucciones para procesar datos de vídeo que, cuando se ejecutan, hacen que uno o más procesadores lleven a cabo el procedimiento según las reivindicaciones 1 a 6.

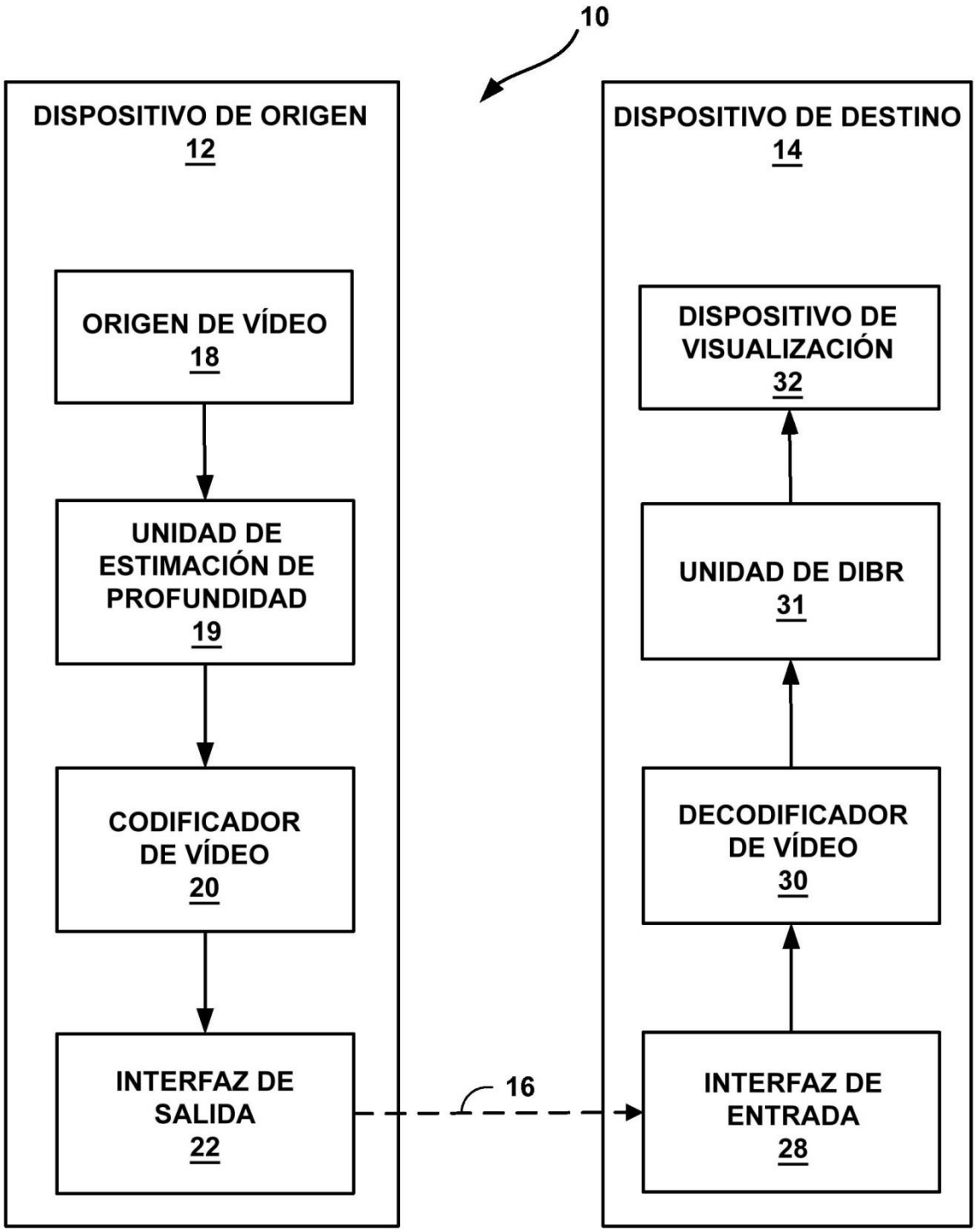
10
DISPOSITIVO DE ORIGEN
12
ORIGEN DE VÍDEO
18
UNIDAD DE ESTIMACIÓN DE PROFUNDIDAD
19
CODIFICADOR DE VÍDEO
20
INTERFAZ DE SALIDA
22
16
DISPOSITIVO DE DESTINO
14
DISPOSITIVO DE VISUALIZACIÓN
32
UNIDAD DE DIBR
31
DECODIFICADOR DE VÍDEO
30
INTERFAZ DE ENTRADA
28

FIG. 1

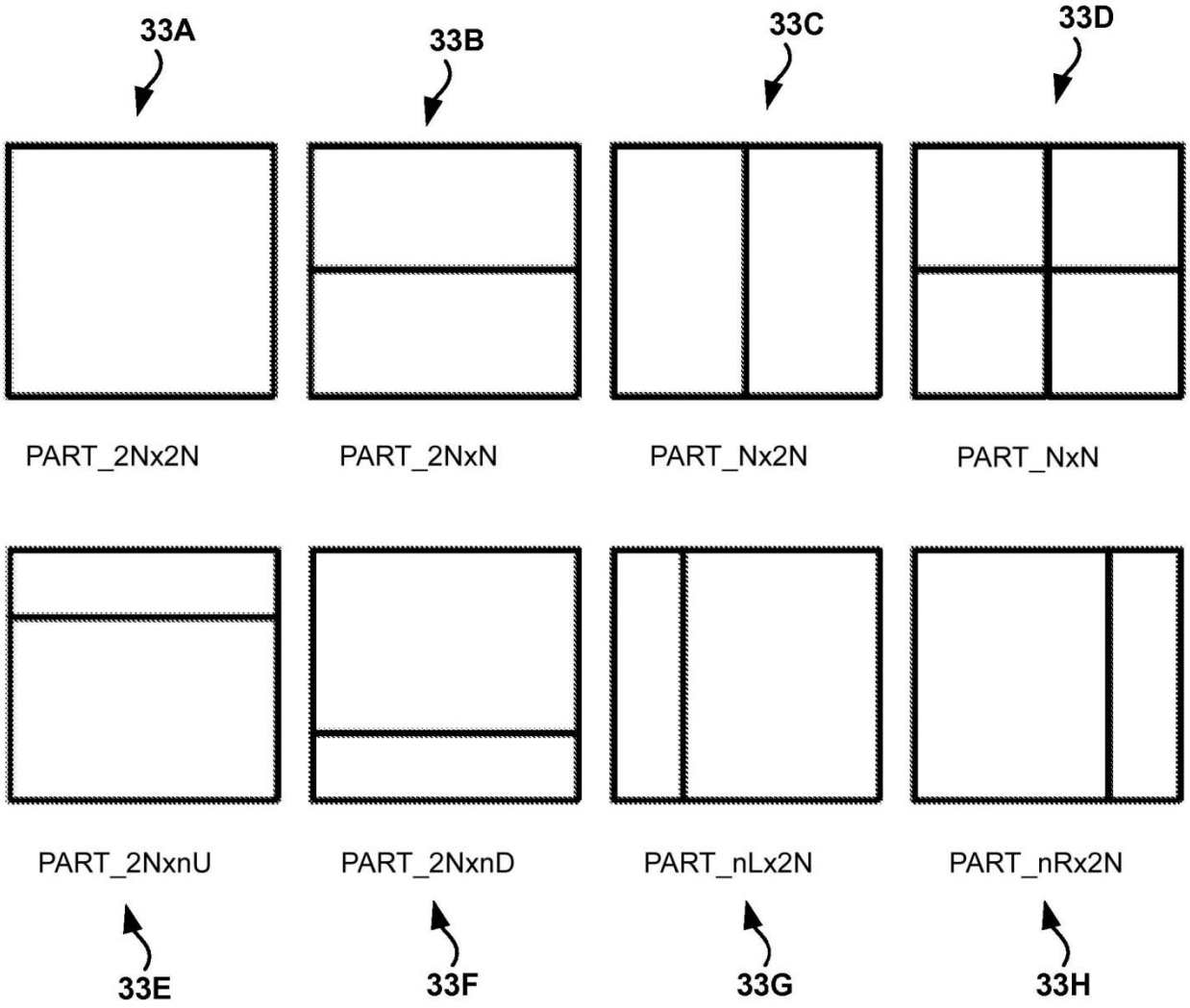
33A
33B
33C
33D
PART_2Nx2N
PART_2NxN
PART_Nx2N
PART_NxN
PART_2NxnU
PART_2NxnD
PART_nLx2N
PART_nRx2N
33E
33F
33G
33H

FIG. 2

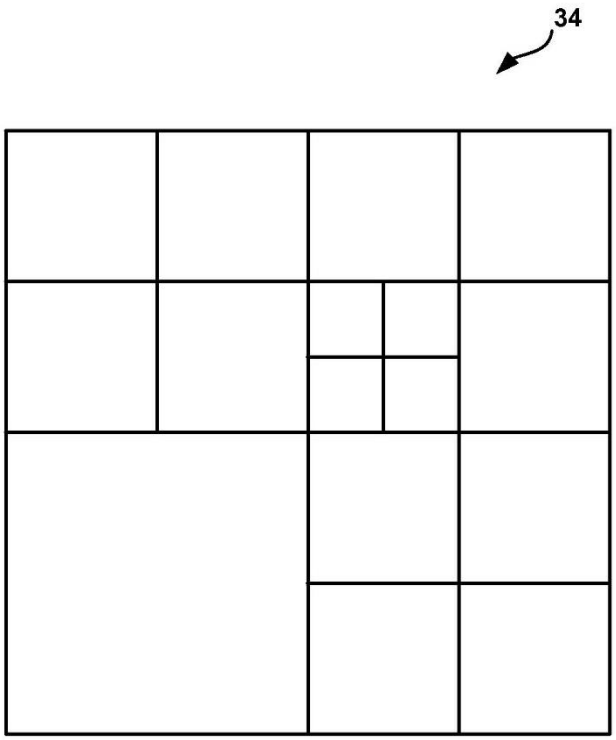

FIG. 3

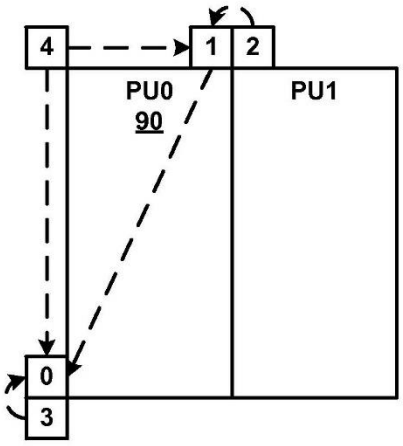

FIG. 4A

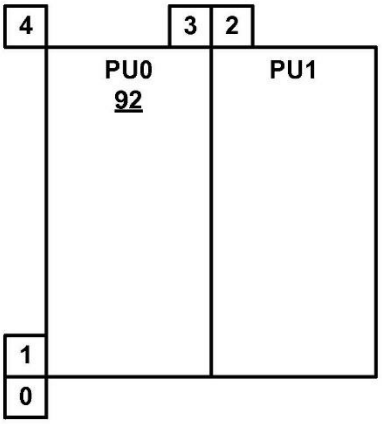

FIG. 4B

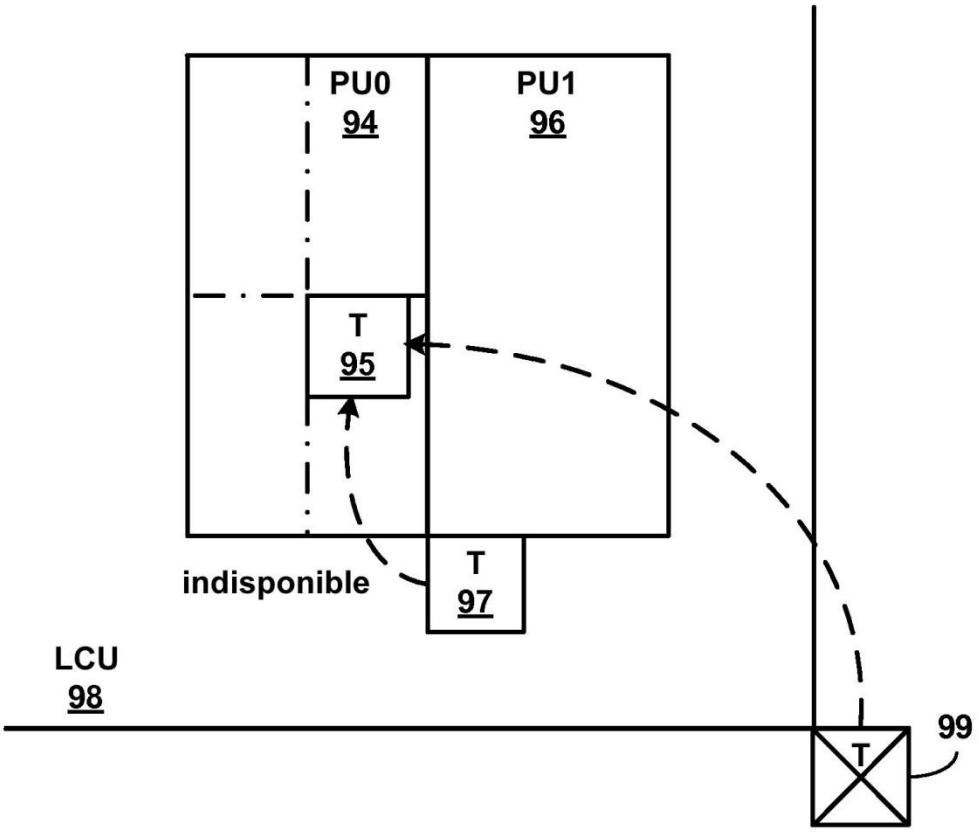

FIG. 5A

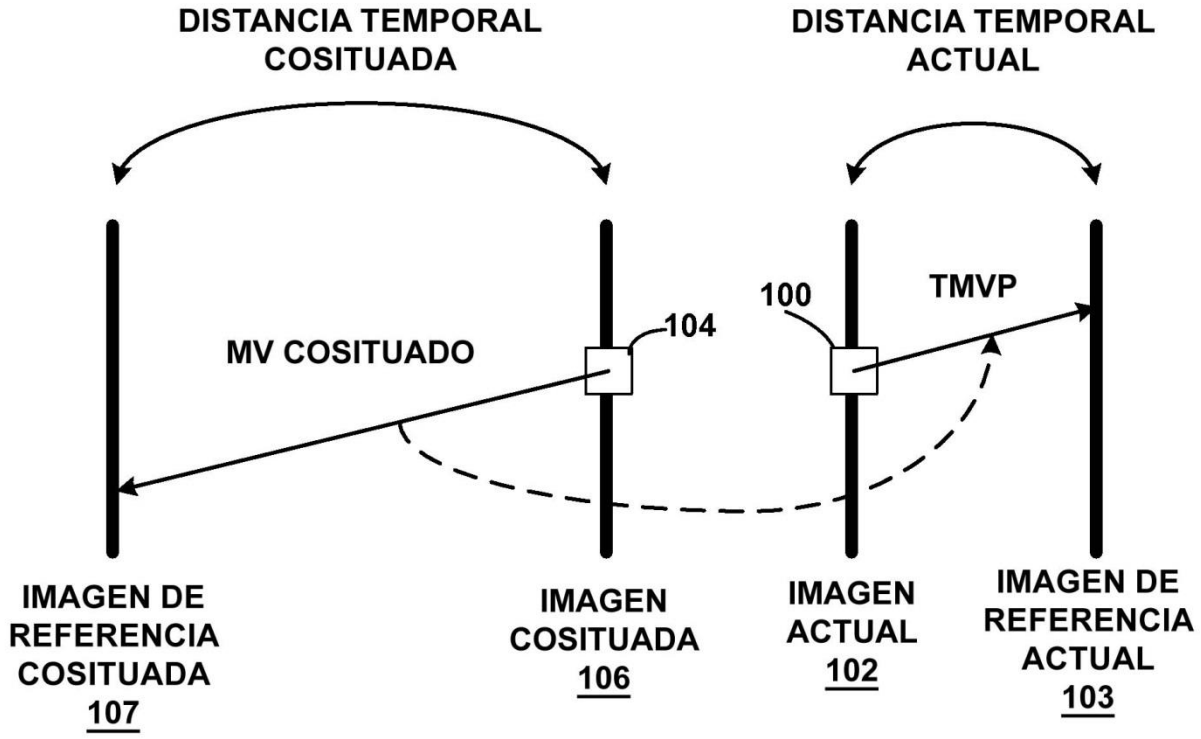
DISTANCIA TEMPORAL COSITUADA
DISTANCIA TEMPORAL ACTUAL
MV COSITUADO
104
100
TMVP
IMAGEN DE REFERENCIA COSITUADA 107
IMAGEN COSITUADA 106
IMAGEN ACTUAL 102
IMAGEN DE REFERENCIA ACTUAL 103

FIG. 5B

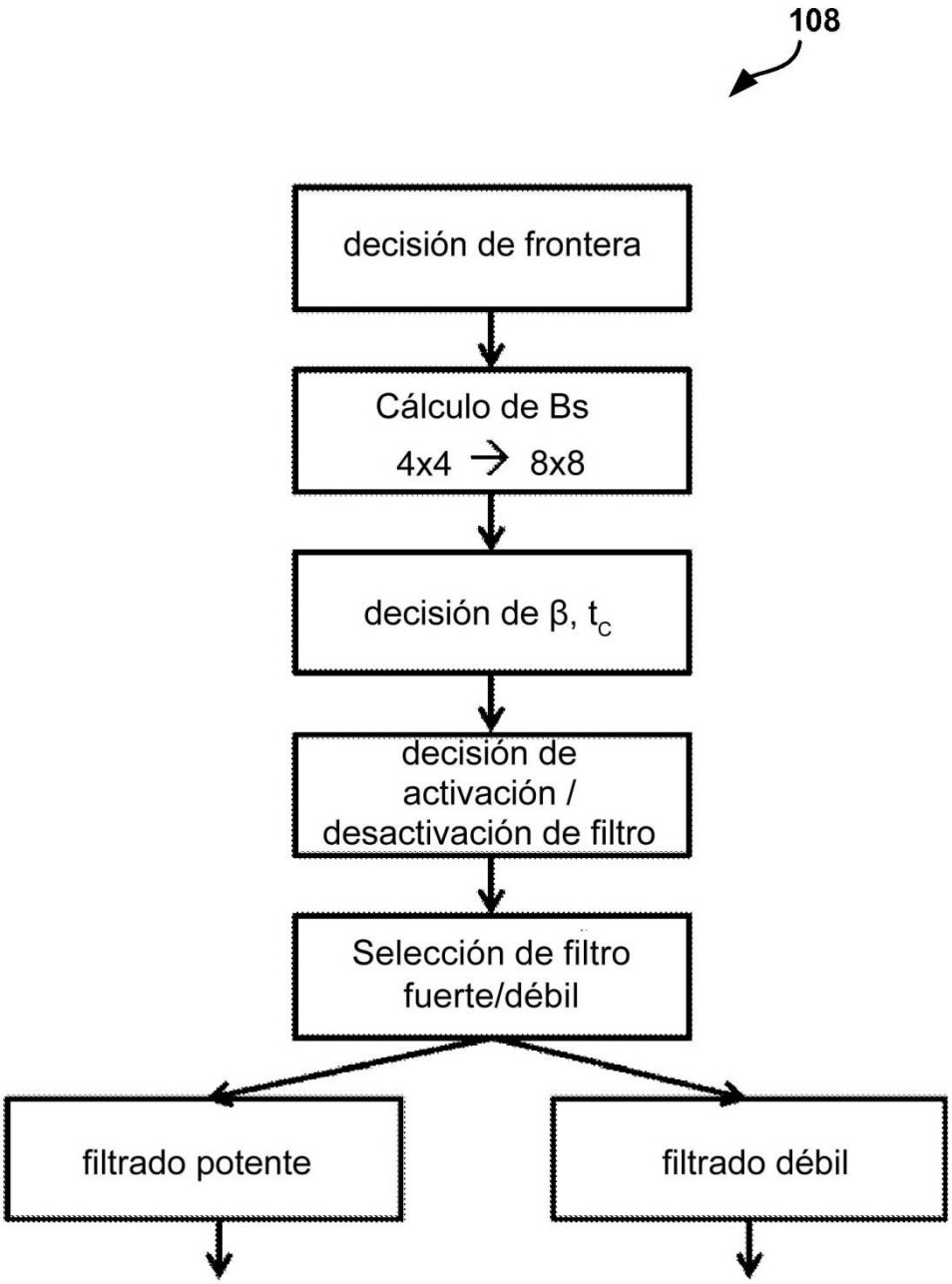
108
decisión de frontera
Cálculo de Bs
4x4 → 8x8
decisión de β, t_C
decisión de activación / desactivación de filtro
Selección de filtro fuerte/débil
filtrado potente
filtrado débil

FIG. 6

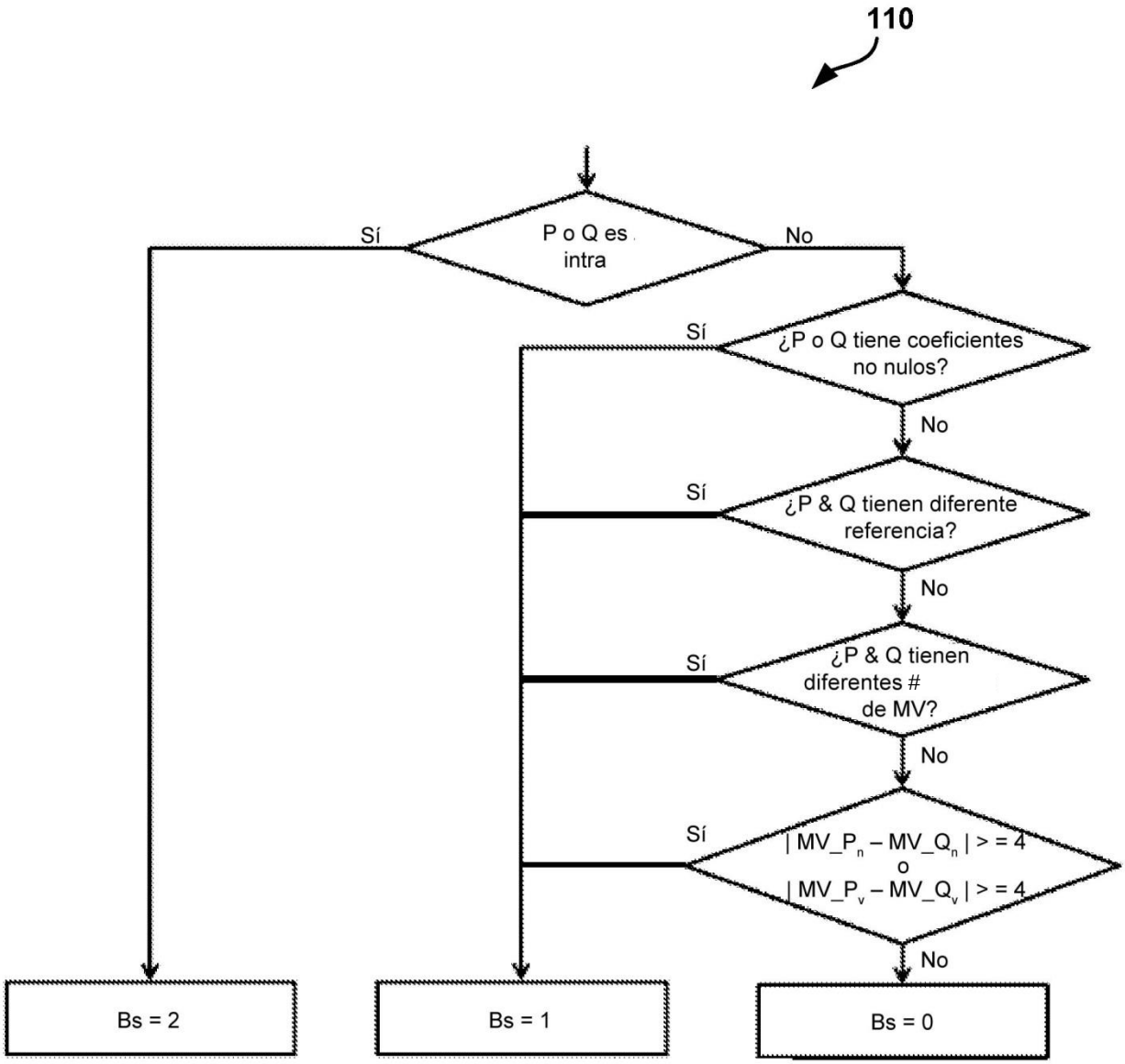
110
P o Q es intra
Sí
No
¿P o Q tiene coeficientes no nulos?
Sí
No
¿P & Q tienen diferente referencia?
Sí
No
¿P & Q tienen diferentes # de MV?
Sí
No
| MV_P_n – MV_Q_n | > = 4
o
| MV_P_v – MV_Q_v | > = 4
Sí
No
Bs = 2
Bs = 1
Bs = 0

FIG. 7

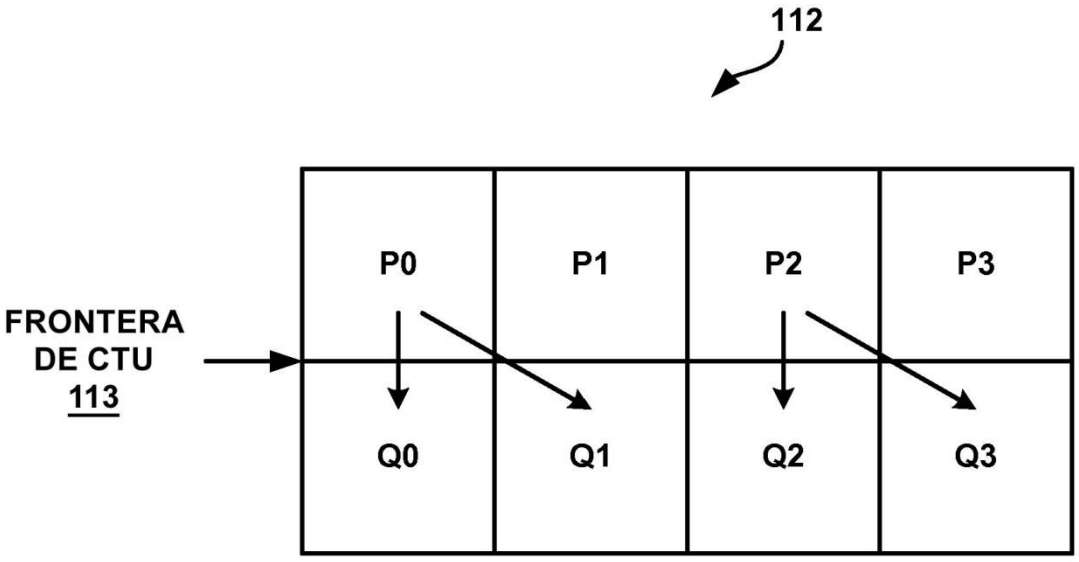

FIG. 8

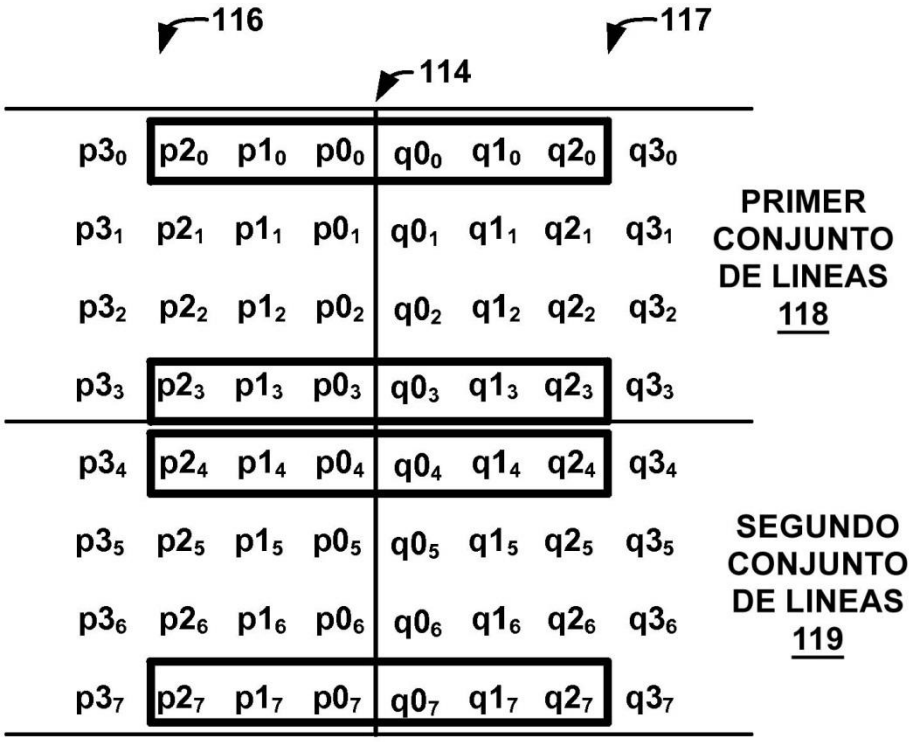
116
117
114
p3₀ p2₀ p1₀ p0₀ q0₀ q1₀ q2₀ q3₀
p3₁ p2₁ p1₁ p0₁ q0₁ q1₁ q2₁ q3₁
p3₂ p2₂ p1₂ p0₂ q0₂ q1₂ q2₂ q3₂
p3₃ p2₃ p1₃ p0₃ q0₃ q1₃ q2₃ q3₃
p3₄ p2₄ p1₄ p0₄ q0₄ q1₄ q2₄ q3₄
p3₅ p2₅ p1₅ p0₅ q0₅ q1₅ q2₅ q3₅
p3₆ p2₆ p1₆ p0₆ q0₆ q1₆ q2₆ q3₆
p3₇ p2₇ p1₇ p0₇ q0₇ q1₇ q2₇ q3₇
PRIMER CONJUNTO DE LINEAS 118
SEGUNDO CONJUNTO DE LINEAS 119

FIG. 9

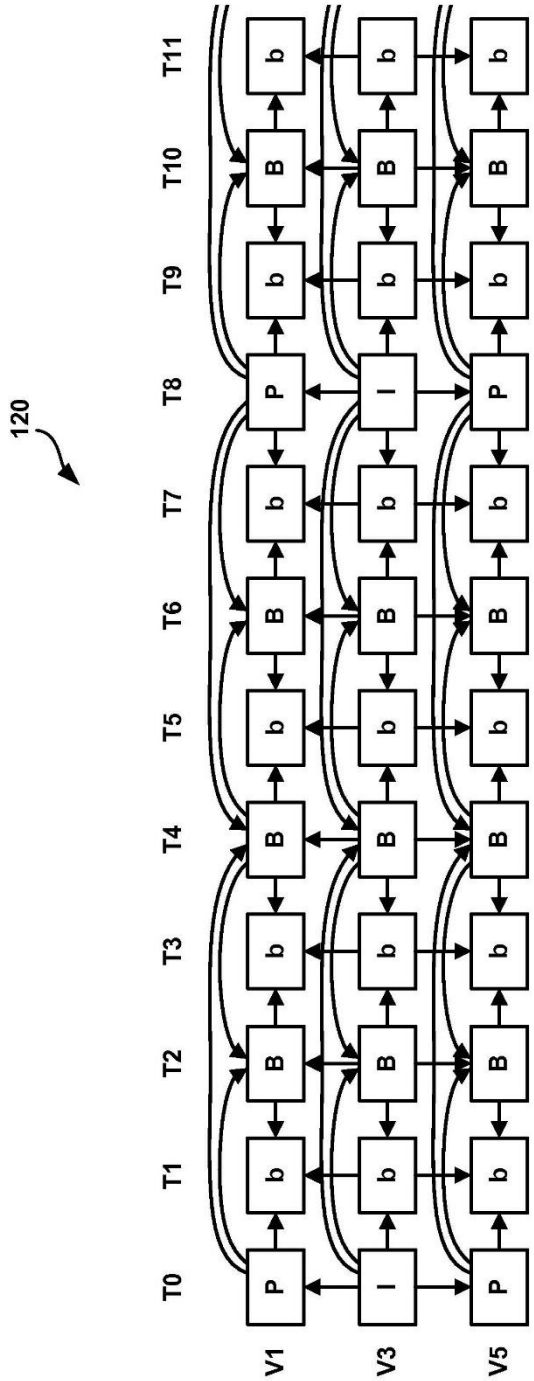

FIG. 10

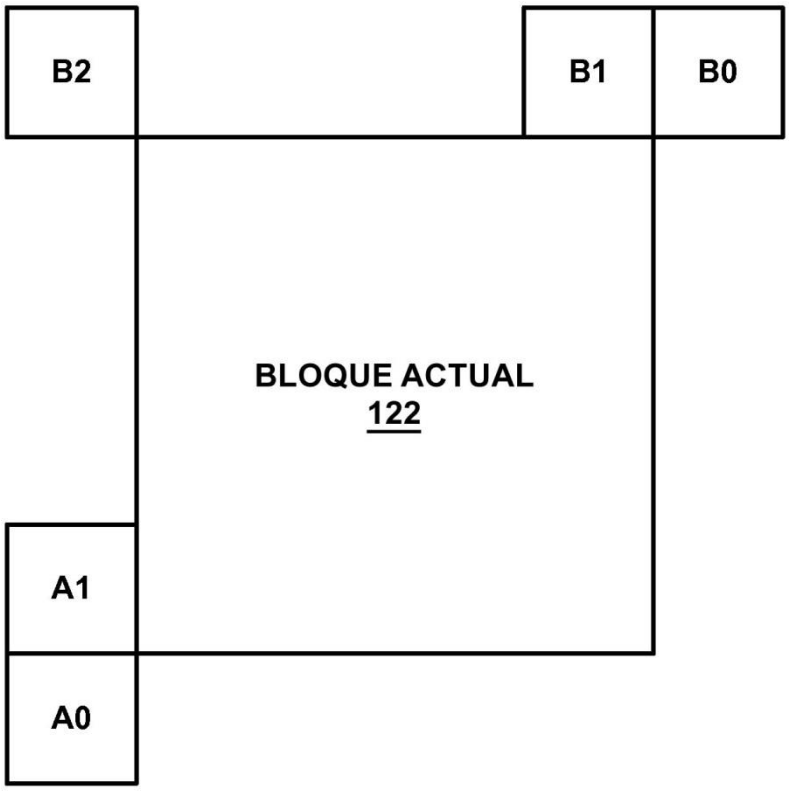

FIG. 11

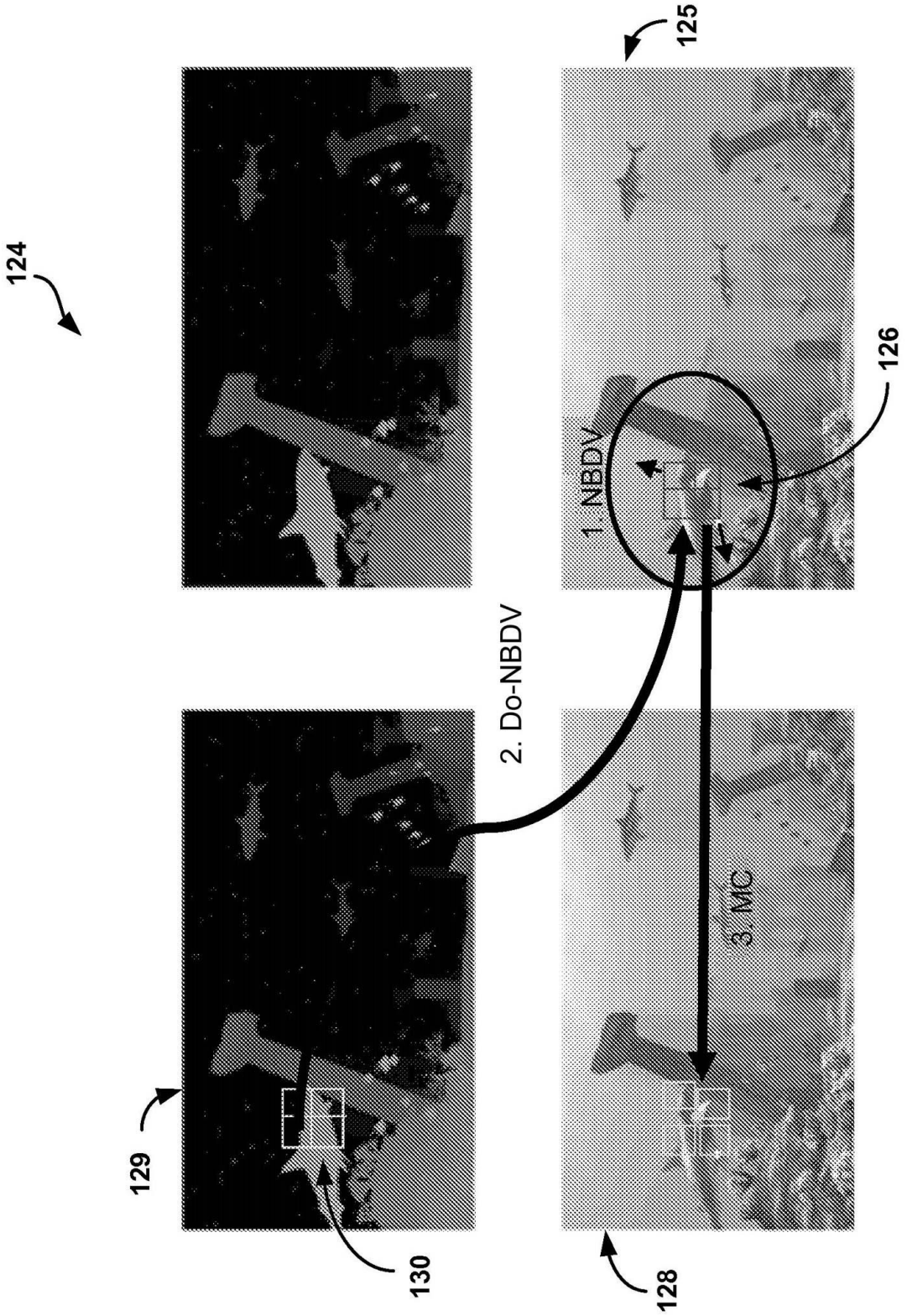

FIG. 12

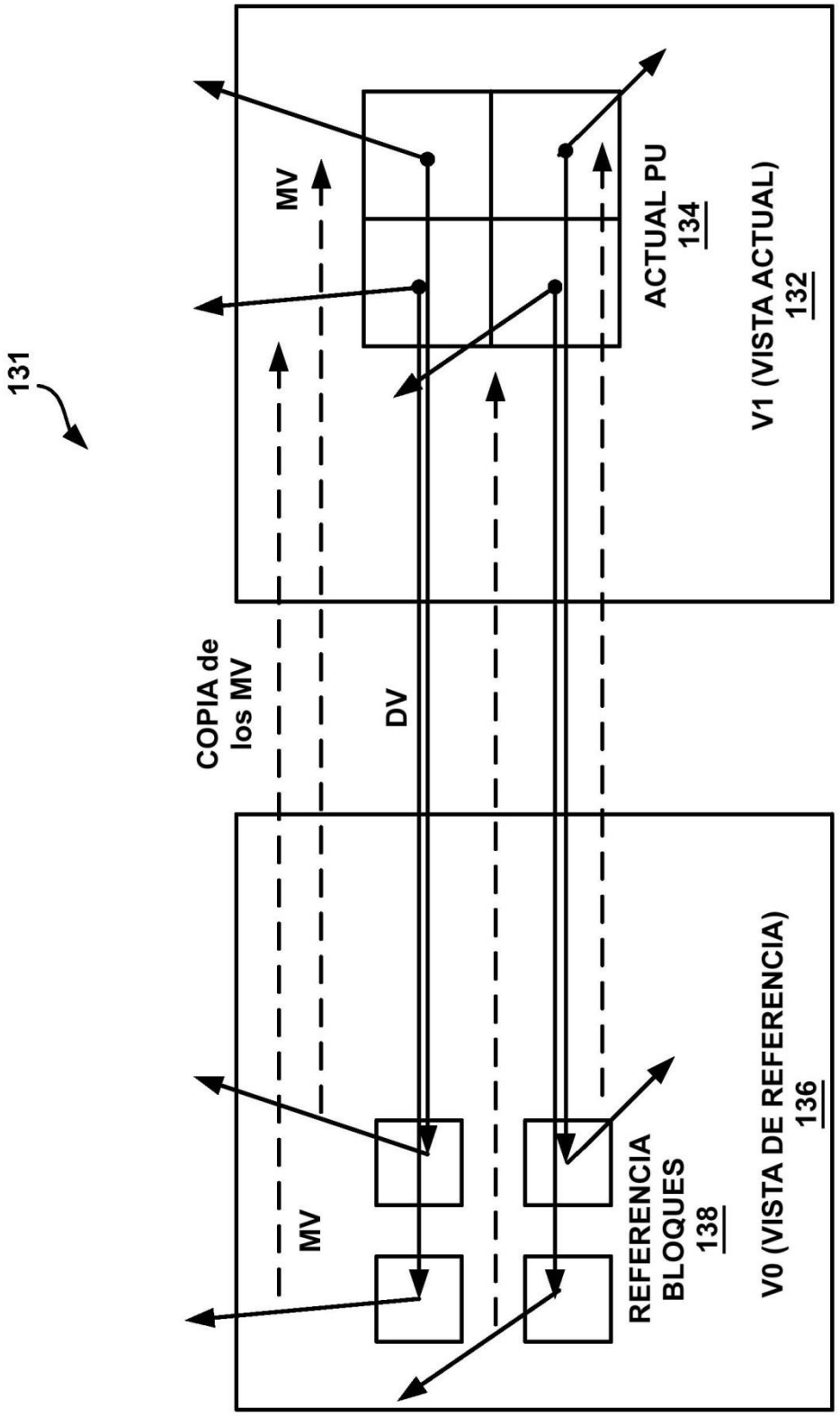
131
MV
ACTUAL PU
134
V1 (VISTA ACTUAL)
132
COPIA de
los MV
DV
MV
REFERENCIA
BLOQUES
138
V0 (VISTA DE REFERENCIA)
136

FIG. 13

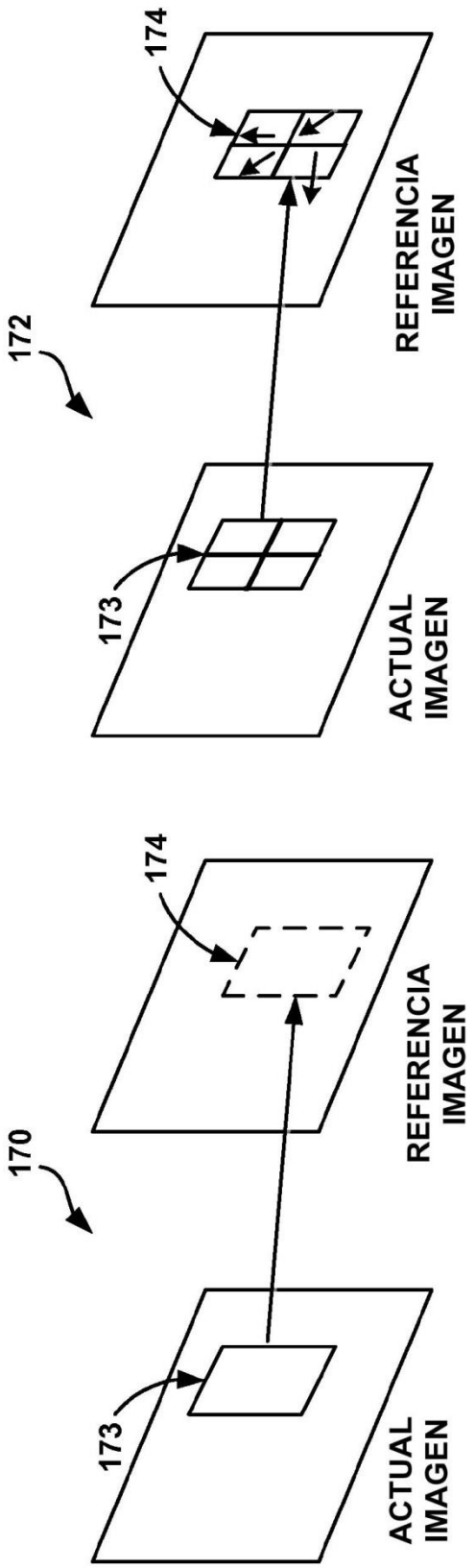

FIG. 14A

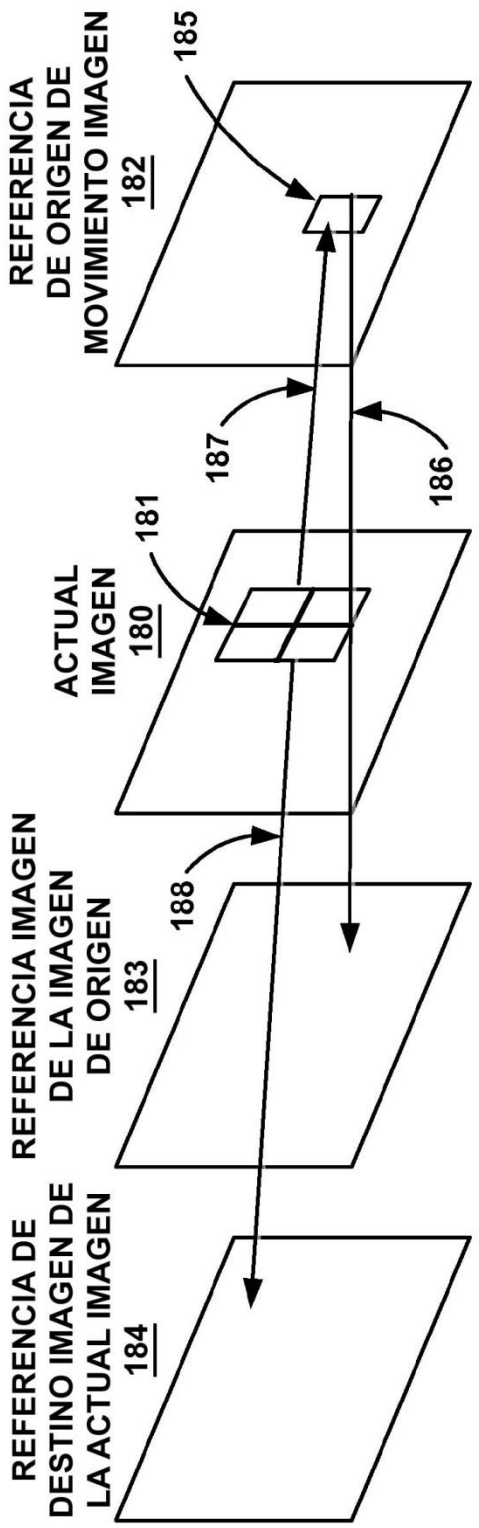

FIG. 14B

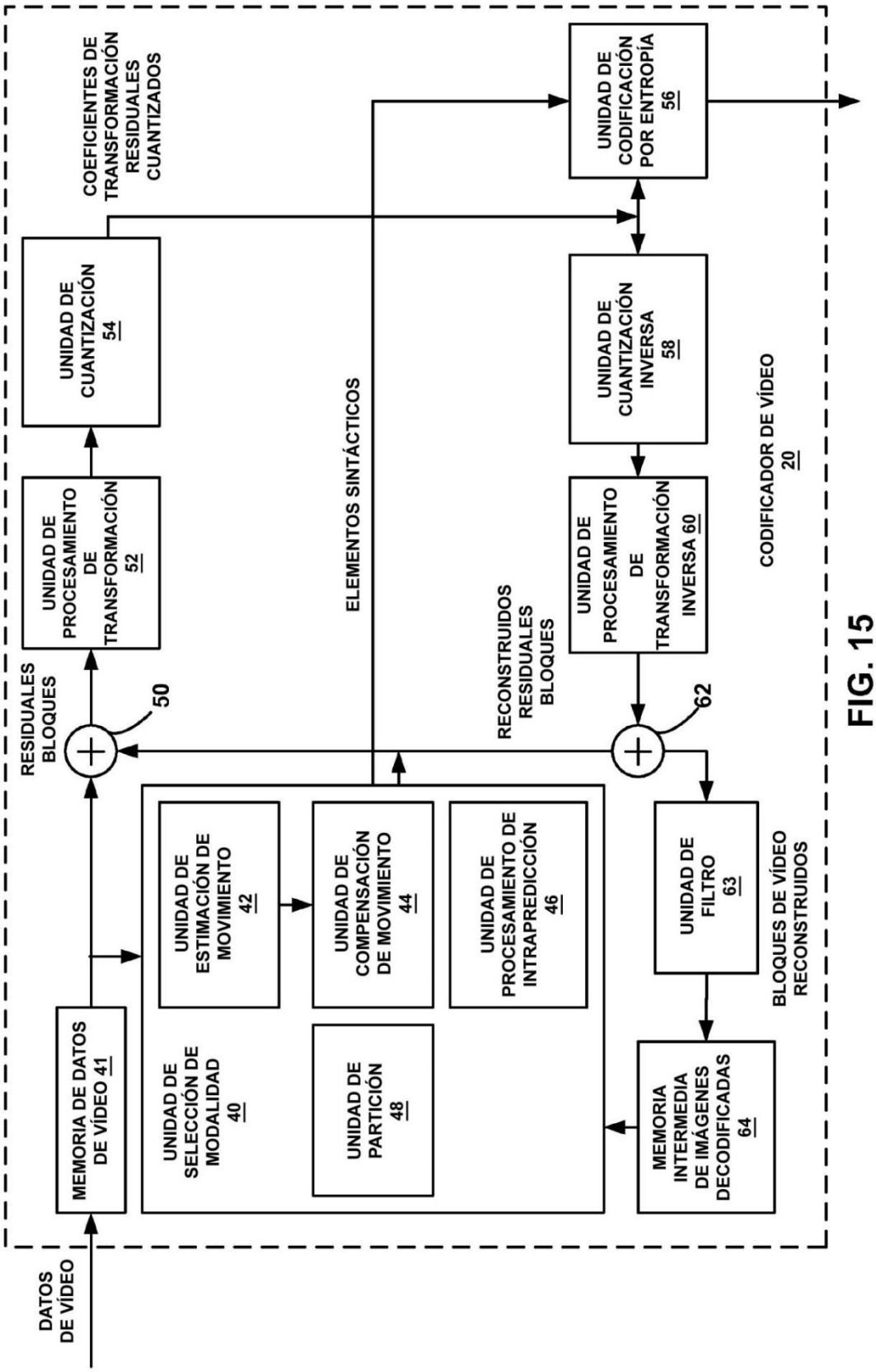
DATOS DE VÍDEO
MEMORIA DE DATOS DE VÍDEO 41
UNIDAD DE SELECCIÓN DE MODALIDAD 40
UNIDAD DE ESTIMACIÓN DE MOVIMIENTO 42
UNIDAD DE COMPENSACIÓN DE MOVIMIENTO 44
UNIDAD DE PROCESAMIENTO DE INTRAPREDICCIÓN 46
UNIDAD DE PARTICIÓN 48
RESIDUALES BLOQUES
50
UNIDAD DE PROCESAMIENTO DE TRANSFORMACIÓN 52
UNIDAD DE CUANTIZACIÓN 54
COEFICIENTES DE TRANSFORMACIÓN RESIDUALES CUANTIZADOS
ELEMENTOS SINTÁCTICOS
UNIDAD DE CODIFICACIÓN POR ENTROPÍA 56
UNIDAD DE CUANTIZACIÓN INVERSA 58
UNIDAD DE PROCESAMIENTO DE TRANSFORMACIÓN INVERSA 60
RECONSTRUIDOS RESIDUALES BLOQUES
62
UNIDAD DE FILTRO 63
BLOQUES DE VÍDEO RECONSTRUIDOS
MEMORIA INTERMEDIA DE IMÁGENES DECODIFICADAS 64
CODIFICADOR DE VÍDEO 20

FIG. 15

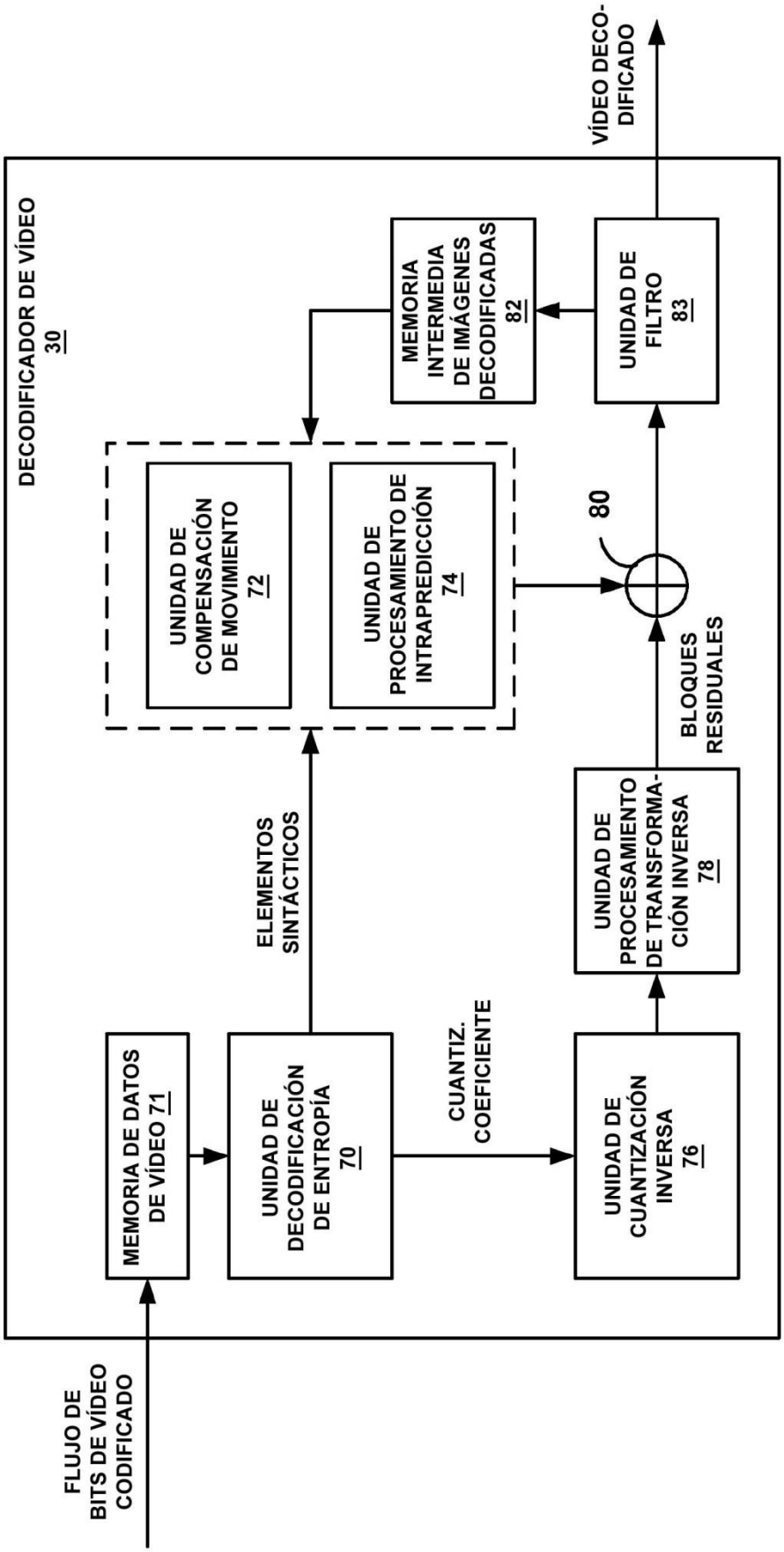
DECODIFICADOR DE VÍDEO
30
FLUJO DE BITS DE VÍDEO CODIFICADO
MEMORIA DE DATOS DE VÍDEO 71
UNIDAD DE DECODIFICACIÓN DE ENTROPÍA 70
ELEMENTOS SINTÁCTICOS
UNIDAD DE COMPENSACIÓN DE MOVIMIENTO 72
UNIDAD DE PROCESAMIENTO DE INTRAPREDICCIÓN 74
MEMORIA INTERMEDIA DE IMÁGENES DECODIFICADAS 82
CUANTIZ. COEFICIENTE
UNIDAD DE CUANTIZACIÓN INVERSA 76
UNIDAD DE PROCESAMIENTO DE TRANSFORMACIÓN INVERSA 78
80
BLOQUES RESIDUALES
UNIDAD DE FILTRO 83
VÍDEO DECODIFICADO

FIG. 16

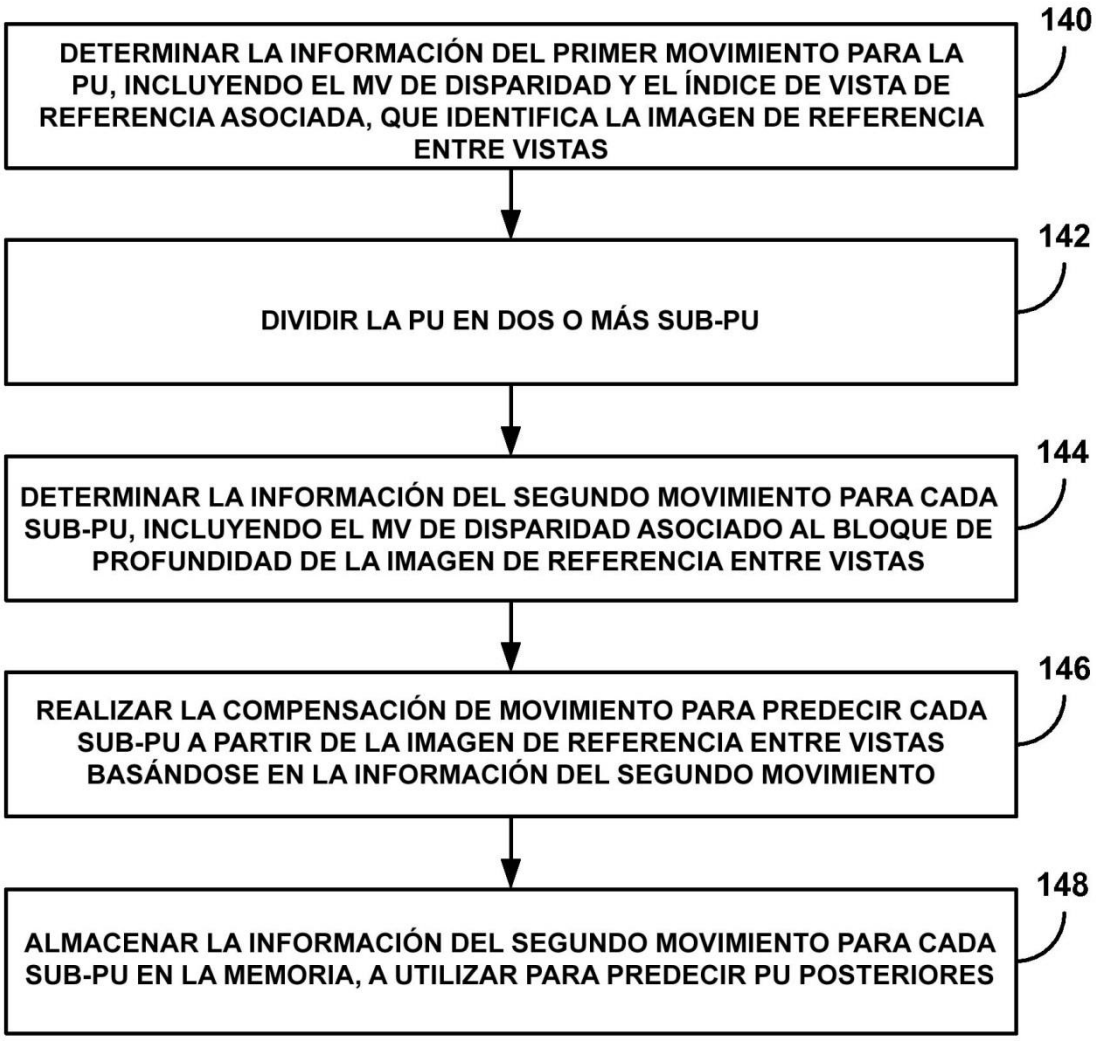
140
DETERMINAR LA INFORMACIÓN DEL PRIMER MOVIMIENTO PARA LA PU, INCLUYENDO EL MV DE DISPARIDAD Y EL ÍNDICE DE VISTA DE REFERENCIA ASOCIADA, QUE IDENTIFICA LA IMAGEN DE REFERENCIA ENTRE VISTAS
142
DIVIDIR LA PU EN DOS O MÁS SUB-PU
144
DETERMINAR LA INFORMACIÓN DEL SEGUNDO MOVIMIENTO PARA CADA SUB-PU, INCLUYENDO EL MV DE DISPARIDAD ASOCIADO AL BLOQUE DE PROFUNDIDAD DE LA IMAGEN DE REFERENCIA ENTRE VISTAS
146
REALIZAR LA COMPENSACIÓN DE MOVIMIENTO PARA PREDECIR CADA SUB-PU A PARTIR DE LA IMAGEN DE REFERENCIA ENTRE VISTAS BASÁNDOSE EN LA INFORMACIÓN DEL SEGUNDO MOVIMIENTO
148
ALMACENAR LA INFORMACIÓN DEL SEGUNDO MOVIMIENTO PARA CADA SUB-PU EN LA MEMORIA, A UTILIZAR PARA PREDECIR PU POSTERIORES

FIG. 17

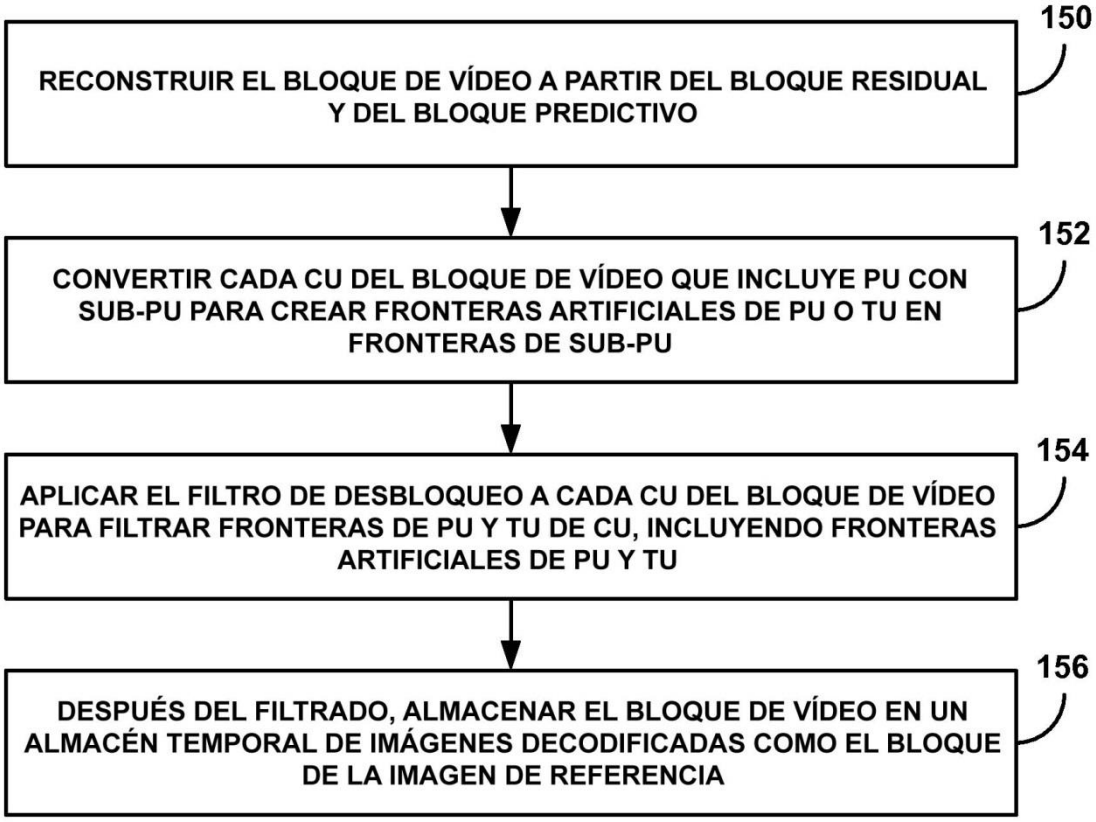
150
RECONSTRUIR EL BLOQUE DE VÍDEO A PARTIR DEL BLOQUE RESIDUAL Y DEL BLOQUE PREDICTIVO
152
CONVERTIR CADA CU DEL BLOQUE DE VÍDEO QUE INCLUYE PU CON SUB-PU PARA CREAR FRONTERAS ARTIFICIALES DE PU O TU EN FRONTERAS DE SUB-PU
154
APLICAR EL FILTRO DE DESBLOQUEO A CADA CU DEL BLOQUE DE VÍDEO PARA FILTRAR FRONTERAS DE PU Y TU DE CU, INCLUYENDO FRONTERAS ARTIFICIALES DE PU Y TU
156
DESPUÉS DEL FILTRADO, ALMACENAR EL BLOQUE DE VÍDEO EN UN ALMACÉN TEMPORAL DE IMÁGENES DECODIFICADAS COMO EL BLOQUE DE LA IMAGEN DE REFERENCIA

FIG. 18

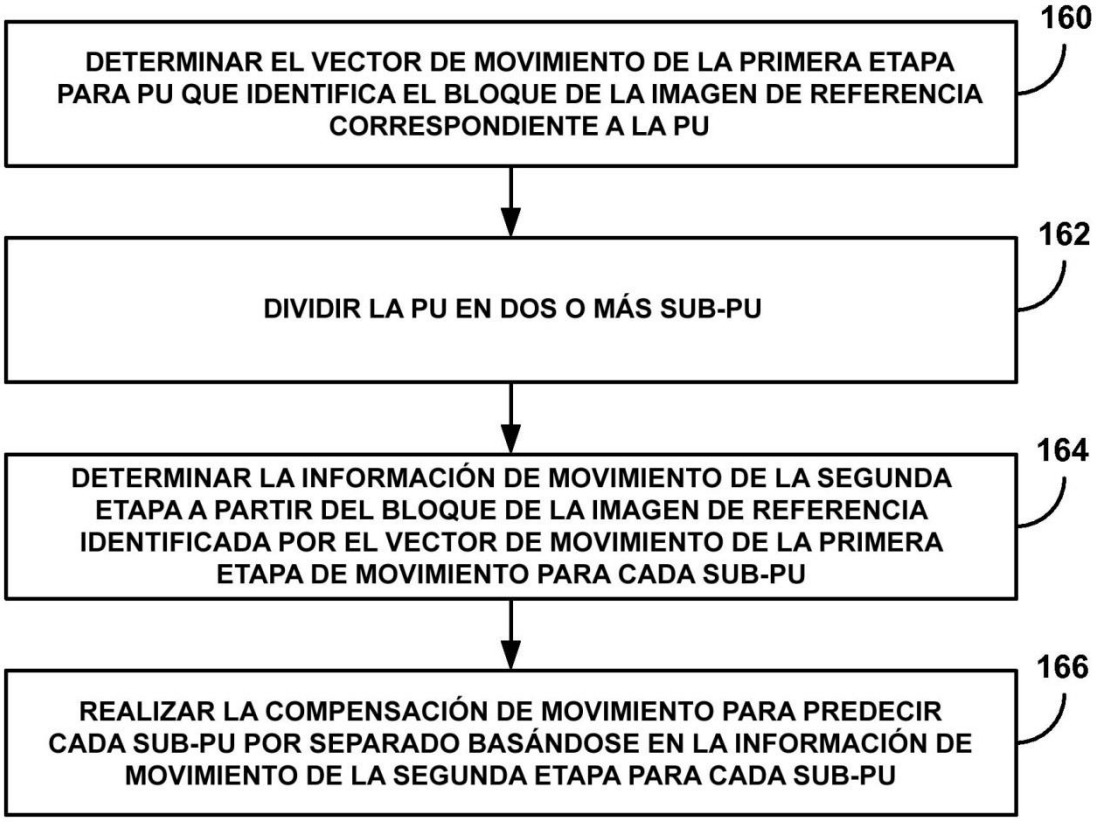
160
DETERMINAR EL VECTOR DE MOVIMIENTO DE LA PRIMERA ETAPA PARA PU QUE IDENTIFICA EL BLOQUE DE LA IMAGEN DE REFERENCIA CORRESPONDIENTE A LA PU
162
DIVIDIR LA PU EN DOS O MÁS SUB-PU
164
DETERMINAR LA INFORMACIÓN DE MOVIMIENTO DE LA SEGUNDA ETAPA A PARTIR DEL BLOQUE DE LA IMAGEN DE REFERENCIA IDENTIFICADA POR EL VECTOR DE MOVIMIENTO DE LA PRIMERA ETAPA DE MOVIMIENTO PARA CADA SUB-PU
166
REALIZAR LA COMPENSACIÓN DE MOVIMIENTO PARA PREDECIR CADA SUB-PU POR SEPARADO BASÁNDOSE EN LA INFORMACIÓN DE MOVIMIENTO DE LA SEGUNDA ETAPA PARA CADA SUB-PU

FIG. 19