

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 842 109**

51 Int. Cl.:

H04N 19/597 (2014.01)

H04N 19/577 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **09.01.2015 PCT/US2015/010878**

87 Fecha y número de publicación internacional: **16.07.2015 WO15106141**

96 Fecha de presentación y número de la solicitud europea: **09.01.2015 E 15703144 (4)**

97 Fecha y número de publicación de la concesión europea: **07.10.2020 EP 3092805**

54 Título: **Predicción residual avanzada basada en bloques para la codificación de vídeo 3D**

30 Prioridad:

11.01.2014 US 201461926290 P
08.01.2015 US 201514592633

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
12.07.2021

73 Titular/es:

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

72 Inventor/es:

ZHANG, LI y
CHEN, YING

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 842 109 T3

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

DESCRIPCIÓN

Predicción residual avanzada basada en bloques para la codificación de vídeo 3D

5 CAMPO TÉCNICO

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

10 ANTECEDENTES

[0002] Las capacidades de vídeo digital se pueden incorporar en una amplia gama de dispositivos, incluyendo 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 continua de vídeo y similares. Los dispositivos de vídeo digital implementan técnicas de codificación de vídeo, tales como las descritas en las normas definidas por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificación Avanzada de Vídeo (AVC), la norma de Codificación de Vídeo de Alta Eficacia (HEVC), actualmente en desarrollo, y las ampliaciones de tales normas. Los dispositivos de vídeo pueden transmitir, recibir, codificar, descodificar y/o almacenar información de vídeo digital más eficazmente, implementando dichas técnicas de codificación de vídeo.

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

[0004] La predicción espacial o temporal da como resultado un bloque predictivo para un bloque que se va a codificar. Los datos residuales representan diferencias de píxeles entre el bloque original que se va a codificar y el bloque predictivo. Un bloque intercodificado se codifica de acuerdo con un vector de movimiento que apunta a un bloque de muestras de referencia que forman el bloque predictivo, y los datos residuales que indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intracodificado se codifica de acuerdo con un modo de intracodificación y los datos residuales. Para una mayor compresión, los datos residuales se pueden transformar desde el dominio de píxel a un dominio de transformada, dando como resultado coeficientes de transformada residuales, que a continuación se pueden cuantificar. Los coeficientes de transformada cuantificados, dispuestos inicialmente en una matriz bidimensional, se pueden explorar para producir un vector unidimensional de coeficientes de transformada, y se puede aplicar la codificación por entropía para lograr aún más compresión.

[0005] En "3D-CE4.h: Simplification of inter-view ARP" de Sugimoto et al (Extensiones de codificación de vídeo 3D de JCT, 7ª reunión, San José, número de documento JCT3V-G158, 9 de enero de 2014) se afirma que 3D-HEVC admite la predicción residual avanzada (ARP) tanto para residual temporal como para residual entrevista y se propone reunir esos diferentes procesos ARP direccionales en un proceso ARP temporal.

[0006] En "CE4: Further improvements on advanced residual prediction" de Zhang et al Extensiones de codificación de vídeo 3D de JCT, 6ª reunión, Ginebra, número de documento JCT3V-F0123, 18 de octubre de 2013), se propone que: ARP se extiende para el residuo entrevista, cuando el bloque actual utiliza la predicción entrevista a partir de una imagen de referencia entrevista.

55 BREVE EXPLICACIÓN

[0007] En general, esta divulgación está relacionada con la codificación de vídeo multivista, en la que los datos de vídeo codificados incluyen dos o más vistas. En particular, esta divulgación describe diversas técnicas relacionadas con la predicción residual avanzada (ARP). Las técnicas de esta divulgación pueden reducir la cantidad de veces que un codificador de vídeo (por ejemplo, codificador de vídeo y/o descodificador de vídeo) accede a información de movimiento para realizar ARP, o cualesquiera procesos de compensación de movimiento subyacentes (es decir, usando vectores de movimiento asignados para generar los bloques de predicción con potenciales funcionamientos de interpolación). De esta manera, la velocidad de codificación de vídeo (es decir, codificación o descodificación) se puede incrementar ya que se realizan menos accesos a memoria para información de movimiento.

[0008] La invención se define mediante las reivindicaciones adjuntas.

[0009] En un ejemplo de la divulgación, un procedimiento para descodificar datos de vídeo comprende recibir un primer bloque codificado de datos de vídeo en una primera unidad de acceso de una primera vista, en el que el primer bloque codificado de datos de vídeo fue codificado usando predicción residual avanzada y predicción bidireccional, incluyendo la predicción bidireccional predicción temporal para una primera dirección de predicción y la predicción entrevista para una segunda dirección de predicción, determinar información de movimiento temporal para la primera dirección de predicción para el primer bloque codificado de datos de vídeo, determinar información de movimiento de disparidad para segunda dirección de predicción para el primer bloque codificado de datos de vídeo, identificar bloques de referencia para la segunda dirección de predicción, diferente a la primera dirección de predicción, usando la información de movimiento temporal determinada para la primera dirección de predicción, en el que los bloques de referencia están en una unidad de acceso diferente de la primera unidad de acceso, y realizar predicción residual avanzada en el primer bloque codificado de datos de vídeo usando los bloques de referencia identificados para la segunda dirección de predicción.

[0010] En otro ejemplo de la divulgación, un aparato configurado para descodificar datos de vídeo comprende una memoria de datos de vídeo configurada para almacenar un primer bloque codificado de datos de vídeo en una primera unidad de acceso de una primera vista, en el que el primer bloque codificado de datos de vídeo fue codificado usando predicción residual avanzada y predicción bidireccional, incluyendo la predicción bidireccional predicción temporal para una primera dirección de predicción y la predicción entrevista para una segunda dirección de predicción, y uno o más procesadores en comunicación con la memoria de datos de vídeo y configurados para determinar información de movimiento para la primera dirección de predicción para el primer bloque codificado de datos de vídeo, determinar información de movimiento de disparidad para la segunda dirección de predicción para el primer bloque codificado de datos de vídeo, identificar bloques de referencia para la segunda dirección de predicción, diferente de la primera dirección de predicción, usando la información de movimiento temporal determinada para la primera dirección de predicción, en el que los bloques de referencia están en una unidad de acceso diferente a la primera unidad de acceso, y realizan la predicción residual avanzada en el primer bloque codificado de datos de vídeo usando los bloques de referencia identificados para la segunda dirección de predicción.

[0011] En otro ejemplo de la divulgación, un aparato configurado para descodificar datos de vídeo comprende medios para recibir un primer bloque codificado de datos de vídeo en una primera unidad de acceso de una primera vista, en el que el primer bloque codificado de datos de vídeo fue codificado usando predicción residual avanzada y predicción bidireccional, incluyendo la predicción bidireccional predicción temporal para una primera dirección de predicción y la predicción entrevista para una segunda dirección de predicción, medios para determinar información de movimiento temporal para la primera dirección de predicción para el primer bloque codificado de datos de vídeo, medios para determinar información de movimiento de disparidad para segunda dirección de predicción para el primer bloque codificado de datos de vídeo, medios para identificar bloques de referencia para la segunda dirección de predicción, diferente a la primera dirección de predicción, usando la información de movimiento temporal determinada para la primera dirección de predicción, en el que los bloques de referencia están en una unidad de acceso diferente de la primera unidad de acceso, y medios para realizar predicción residual avanzada en el primer bloque codificado de datos de vídeo usando los bloques de referencia identificados para la segunda dirección de predicción.

[0012] En otro ejemplo, esta divulgación describe un medio de almacenamiento legible por ordenador que almacena instrucciones que, cuando se ejecutan, hacen que uno o más procesadores de un dispositivo configurado para descodificar datos de vídeo reciban un primer bloque codificado de datos de vídeo en una primera unidad de acceso de una primera vista, en el que el primer bloque codificado de datos de vídeo fue codificado usando predicción residual avanzada y predicción bidireccional, incluyendo la predicción bidireccional predicción temporal para una primera dirección de predicción y la predicción entrevista para una segunda dirección de predicción, determinen información de movimiento temporal para la primera dirección de predicción para el primer bloque codificado de datos de vídeo, determinen información de movimiento de disparidad para segunda dirección de predicción para el primer bloque codificado de datos de vídeo, identifiquen bloques de referencia para la primera dirección de predicción usando la información de movimiento temporal determinada, identifiquen bloques de referencia para la segunda dirección de predicción, diferente a la primera dirección de predicción, usando la información de movimiento temporal determinada para la primera dirección de predicción, en el que los bloques de referencia están en una unidad de acceso diferente de la primera unidad de acceso, y realicen predicción residual avanzada en el primer bloque codificado de datos de vídeo usando los bloques de referencia identificados para la segunda dirección de predicción.

[0013] Los detalles de uno o más ejemplos de la divulgación se exponen en los dibujos adjuntos y la descripción a continuación. Otros rasgos característicos, objetivos y ventajas resultarán evidentes a partir de la descripción, los dibujos y las reivindicaciones.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

[0014]

La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y descodificación de vídeo de ejemplo que puede utilizar las técnicas descritas en esta divulgación.

La FIG. 2 es un diagrama gráfico que ilustra un orden de codificación o decodificación multivista de ejemplo.

La FIG. 3 es un diagrama conceptual que ilustra un patrón de predicción temporal e entrevista para la codificación de vídeo multivista de ejemplo.

La FIG. 4 es un diagrama conceptual que ilustra una valores de profundidad y textura para vídeo 3D.

La FIG. 5 es un diagrama conceptual que ilustra una relación de bloques contiguos con un bloque actual para predicción de la información de movimiento del bloque actual de ejemplo.

La FIG. 6 es un diagrama conceptual que ilustra un ejemplo de derivación de un candidato de vector de movimiento predicho entrevista y un candidato de vector de movimiento de disparidad entrevista para la predicción de la información de movimiento del bloque actual.

La FIG. 7 es un diagrama conceptual que ilustra bloques contiguos espaciales de ejemplo, a partir de los cuales se puede obtener un vector de disparidad para un bloque de vídeo actual usando la obtención de vector de disparidad basado en bloques contiguos (NBDV), en relación con el bloque de vídeo actual.

La FIG. 8 es un diagrama conceptual que ilustra la predicción de movimiento entrevista de sub-unidad de predicción (PU)

La FIG. 9 es un diagrama conceptual que ilustra un ejemplo de estructura de predicción para la Predicción Residual Avanzada (ARP) temporal de un bloque de vídeo predicho temporalmente.

La FIG. 10 es un diagrama conceptual que ilustra un ejemplo de estructura de predicción bidireccional para ARP temporal.

La FIG. 11 es un diagrama conceptual que ilustra un ejemplo de estructura de predicción para ARP entrevista de un bloque de vídeo predicho entrevista de acuerdo con las técnicas descritas en esta divulgación.

La FIG. 12 es un diagrama conceptual que ilustra un ejemplo de estructura de predicción para ARP bidireccional usando predicción entrevista para una lista de imágenes de referencia y predicción temporal para otra lista de imágenes de referencia.

La FIG. 13 es un diagrama conceptual que ilustra un ejemplo de estructura de predicción para ARP bidireccional usando predicción entrevista para una lista de imágenes de referencia y predicción temporal para otra lista de imágenes de referencia de acuerdo con las técnicas de esta divulgación.

La FIG. 14 es un diagrama conceptual que ilustra ARP temporal basada en bloques.

La FIG. 15 es un diagrama conceptual que ilustra ARP entrevista basada en bloques.

La FIG. 16 es un diagrama conceptual que ilustra ARP basada en bloques con un candidato de fusión sub-PU.

La FIG. 17 es un diagrama de bloques que ilustra un codificador de vídeo de ejemplo que puede implementar las técnicas descritas en esta divulgación.

La FIG. 18 es un diagrama de bloques que ilustra un decodificador de vídeo de ejemplo que puede usar las técnicas descritas en esta divulgación.

La FIG. 19 es un diagrama de flujo que ilustra un ejemplo de procedimiento de ARP para codificar un bloque de vídeo de acuerdo con las técnicas descritas en esta divulgación.

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

DESCRIPCIÓN DETALLADA

[0015] En general, esta divulgación está relacionada con la codificación de vídeo multivista, en la que los datos de vídeo codificados incluyen dos o más vistas. En algunos ejemplos, la codificación de vídeo multivista incluye un proceso de codificación de vídeo multivista más profundidad. En algunos ejemplos, la codificación multivista puede incluir codificación de vídeo tridimensional (3D), y se puede denominar codificación de vídeo 3D. En diversos ejemplos de esta divulgación, se describen técnicas para la predicción residual avanzada (APR) en una vista no base de una multivista y/o secuencias de codificación de vídeo 3D. Las técnicas de esta divulgación pueden reducir la cantidad de

veces que un codificador de vídeo (por ejemplo, codificador de vídeo y/o descodificador de vídeo) accede a información de movimiento, por ejemplo, desde la memoria, para realizar ARP, o cualquier interpretación subyacente (por ejemplo, temporal y/o interpretación y bipredicción entrevista). De esta manera, la velocidad de codificación de vídeo (es decir, codificación o descodificación) se puede incrementar ya que se realizan menos accesos a memoria para información de movimiento.

[0016] Por ejemplo, esta divulgación describe un procedimiento para descodificar datos de vídeo que comprende recibir un primer bloque codificado de datos de vídeo en una primera unidad de acceso, en el que el primer bloque codificado de datos de vídeo fue codificado usando predicción residual avanzada y predicción entrevista bidireccional, determinar la información de movimiento temporal para una primera dirección de predicción para el primer bloque codificado de datos de vídeo, e identificar bloques de referencia para una segunda dirección de predicción, diferente de la primera dirección de predicción, usando la información de movimiento temporal determinada para la primera dirección de predicción, en el que los bloques de referencia están en una segunda unidad de acceso.

[0017] La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y descodificación de vídeo 10 de ejemplo que puede utilizar técnicas de esta divulgación. Como se muestra en la FIG. 1, el sistema 10 incluye un dispositivo de origen 12 que proporciona datos de vídeo codificados que se van a descodificar en un momento posterior por un dispositivo de destino 14. En particular, el dispositivo de origen 12 proporciona los datos de vídeo al dispositivo de destino 14 por medio de un medio legible por ordenador 16. El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender cualquiera de una amplia gama de dispositivos, incluyendo ordenadores de escritorio, ordenadores tipo notebook (es decir, portátiles), tabletas electrónicas, descodificadores, equipos telefónicos tales como los denominados teléfonos "inteligentes", los denominados paneles "inteligentes", televisiones, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de transmisión continua de vídeo o similares. En algunos casos, el dispositivo de origen 12 y el dispositivo de destino 14 pueden estar equipados para la comunicación inalámbrica.

[0018] El dispositivo de destino 14 puede recibir los datos de vídeo codificados que se van a descodificar por medio del medio legible por ordenador 16. El medio legible por ordenador 16 puede comprender cualquier tipo de medio o dispositivo que puede transportar los datos de vídeo codificados desde el dispositivo de origen 12 al dispositivo de destino 14. En un ejemplo, el medio legible por ordenador 16 puede comprender un medio de comunicación para habilitar 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 con una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitir al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrica o por cable, tal como un espectro de radiofrecuencia (RF) o una o más líneas físicas de transmisión. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área amplia o una red global, tal como Internet. El medio de comunicación puede incluir encaminadores, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo de origen 12 al dispositivo de destino 14.

[0019] En algunos ejemplos, se pueden emitir 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 de una variedad de medios de almacenamiento de datos, distribuidos o de acceso local, tales como un disco duro, discos Blu-ray, DVD, CD-ROM, memoria flash, memoria volátil o no volátil, o cualquier otro medio de almacenamiento digital adecuado para almacenar datos de vídeo codificados. En un ejemplo adicional, el dispositivo de almacenamiento puede corresponder a un servidor de ficheros o a otro dispositivo de almacenamiento intermedio que pueda almacenar el vídeo codificado generado por el dispositivo de origen 12. El dispositivo de destino 14 puede acceder a datos de vídeo almacenados desde el dispositivo de almacenamiento por medio de transmisión continua o descarga. El servidor de archivos puede ser cualquier tipo de servidor que pueda almacenar datos de vídeo codificados y transmitir esos datos de vídeo codificados al dispositivo de destino 14. Los servidores de archivos de ejemplo incluyen un servidor web (por ejemplo, para un sitio web), un servidor FTP, dispositivos de almacenamiento conectados en red (NAS) o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de cualquier conexión de datos estándar, incluyendo una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión Wi-Fi), una conexión cableada (por ejemplo, DSL, módem de cable, etc.) o una combinación de ambas cosas que sea adecuada para acceder a datos de vídeo codificados almacenados en un servidor de archivos. La transmisión de datos de vídeo codificados desde el dispositivo de almacenamiento puede ser una transmisión continua, una transmisión de descarga o una combinación de ambas.

[0020] Las técnicas de la presente divulgación no están limitadas necesariamente a aplicaciones o contextos inalámbricos. Las técnicas se pueden aplicar a la codificación de vídeo como soporte de cualquiera de una variedad de aplicaciones de multimedia, tales como radiodifusiones de televisión por el aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones continua de vídeo por internet, tales como la transmisión continua adaptativa dinámica por HTTP (DASH), vídeo digital que se codifica en un medio de almacenamiento de datos, descodificación de vídeo digital almacenado en un medio de almacenamiento de datos u otras aplicaciones. En algunos ejemplos, el sistema 10 puede estar configurado para admitir una transmisión de vídeo unidireccional o

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

[0021] En el ejemplo de la FIG. 1, el dispositivo de origen 12 incluye una fuente 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 una fuente de vídeo externa 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.

[0022] El sistema 10 ilustrado de la FIG. 1 es simplemente un ejemplo. Las técnicas de esta divulgación se pueden realizar mediante cualquier dispositivo de codificación y/o decodificación de vídeo digital. Aunque, en general, las técnicas de esta divulgación se realizan mediante un dispositivo de codificación de vídeo, las técnicas también se pueden realizar mediante un codificador/decodificador de vídeo, denominado típicamente "CÓDEC". Además, las técnicas de esta divulgación también se pueden realizar mediante un preprocesador de vídeo. El dispositivo de origen 12 y el dispositivo de destino 14 son simplemente ejemplos de dichos dispositivos de codificación en los que el dispositivo de origen 12 genera datos de vídeo codificados para su transmisión al dispositivo de destino 14. En algunos ejemplos, los dispositivos 12, 14 pueden funcionar de manera esencialmente simétrica, de modo que cada uno de los dispositivos 12, 14 incluya componentes de codificación y decodificación de vídeo. Por consiguiente, el sistema 10 pueda admitir la transmisión de vídeo unidireccional o bidireccional entre los dispositivos de vídeo 12, 14, por ejemplo, para transmisión continua de vídeo, reproducción de vídeo, radiodifusión de vídeo o videotelefonía.

[0023] La fuente de vídeo 18 del dispositivo de origen 12 puede incluir un dispositivo de captura de vídeo, tal como una videocámara, un archivo de vídeo que contiene vídeo captado previamente y/o una interfaz de alimentación de vídeo para recibir vídeo desde un proveedor de contenidos de vídeo. Como otra alternativa, la fuente de vídeo 18 puede generar datos basados en gráficos de ordenador como el vídeo de origen, o una combinación de vídeo en directo, vídeo archivado y vídeo generado por ordenador. En algunos casos, si la fuente de vídeo 18 es una cámara de vídeo, el dispositivo de origen 12 y el dispositivo de destino 14 pueden formar los llamados teléfonos con cámara o videoteléfonos. Sin embargo, como se menciona anteriormente, las técnicas descritas en esta divulgación pueden ser aplicables a la codificación de vídeo en general, y se pueden aplicar a aplicaciones inalámbricas y/o alámbricas. En cada caso, el codificador de vídeo 20 puede codificar el vídeo capturado, precapturado o generado por ordenador. Después, la información de vídeo codificada se puede emitir por la interfaz de salida 22 a un medio legible por ordenador 16.

[0024] La fuente de vídeo 18 puede proporcionar una o más vistas de datos de vídeo al codificador de vídeo 20. Por ejemplo, la fuente de vídeo 18 puede corresponder a una formación de cámaras, teniendo cada una una posición horizontal única con respecto a una escena particular que se está filmando. De forma alternativa, la fuente de vídeo 18 puede generar datos de vídeo desde perspectivas de cámara horizontal dispares, por ejemplo, usando gráficos por ordenador. La unidad de estimación de profundidad 19 se puede configurar 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.

[0025] Adicionalmente o de forma alternativa, la unidad de estimación de profundidad 19 se puede configurar para calcular valores de profundidad de manera indirecta comparando dos o más imágenes que se capturaron sustancialmente al mismo tiempo desde diferentes perspectivas de cámara horizontal. Calculando la disparidad horizontal entre valores de píxel esencialmente similares en las imágenes, la unidad de estimación de profundidad 19 se puede aproximar a la profundidad de diversos objetos en la escena. La unidad de estimación de profundidad 19 puede estar integrada funcionalmente con la fuente de vídeo 18, en algunos ejemplos. Por ejemplo, cuando la fuente 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.

[0026] El medio legible por ordenador 16 puede incluir medios transitorios, tales como una radiodifusión inalámbrica o 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, por medio de transmisión por red. De forma similar, un dispositivo informático de una instalación de producción de un medio, tal como una instalación de acuñación 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.

[0027] 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 se usa por el descodificador 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 descodificados a un usuario, y puede comprender cualquiera de una variedad de dispositivos de visualización, tales como un tubo de rayos catódicos (CRT), una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de visualización. 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 3D para un espectador.

[0028] La unidad de DIBR 31 del dispositivo de destino 14 puede representar vistas sintetizadas usando información de textura y profundidad de vistas descodificadas, recibidas desde el descodificador de vídeo 30. Por ejemplo, la unidad de DIBR 31 puede determinar la disparidad horizontal para datos de píxeles de imágenes de textura como una función de los valores de píxeles en mapas de profundidad correspondientes. La unidad de DIBR 31 puede generar entonces una imagen sintetizada mediante el desplazamiento de píxeles en una imagen de textura, a la izquierda o a la derecha, en la disparidad horizontal determinada. De esta manera, el dispositivo de visualización 32 puede mostrar una o más vistas, que pueden corresponder a vistas descodificadas y/o vistas sintetizadas, en cualquier combinación. De acuerdo con las técnicas de esta divulgación, el descodificador de vídeo 30 puede proporcionar valores de precisión originales y actualizados para los rangos de profundidad y parámetros de la cámara a la unidad DIBR 31, que puede usar los rangos de profundidad y los parámetros de la cámara para sintetizar adecuadamente las vistas.

[0029] Aunque no se muestra en la FIG. 1, en algunos aspectos, el codificador de vídeo 20 y el descodificador de vídeo 30 se pueden integrar, cada uno, en un codificador y descodificador de audio, y pueden incluir unidades MUX-DEMUX apropiadas, u otro hardware y software, para tratar la codificación tanto de audio como de vídeo en un flujo de datos común o en flujos de datos separados. Si fuera aplicable, las unidades MUX-DEMUX pueden estar conformes con el protocolo multiplexor H.223 de UIT u otros protocolos tales como el protocolo de datagramas de usuario (UDP).

[0030] El codificador de vídeo 20 y el descodificador de vídeo 30 se pueden implementar, cada uno, como cualquiera de una variedad de circuitos codificadores adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), matrices de puertas programables in situ (FPGA), lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. Cuando las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio adecuado no transitorio legible por ordenador y ejecutar las instrucciones en hardware usando uno o más procesadores para realizar las técnicas de esta divulgación. Tanto el codificador de vídeo 20 como el descodificador de vídeo 30 se pueden incluir en uno o más codificadores o descodificadores, cualquiera de los cuales se puede integrar como parte de un codificador/descodificador (CÓDEC) combinado en un dispositivo respectivo. Un dispositivo que incluye el codificador de vídeo 20 y/o el descodificador de vídeo 30 puede comprender un circuito integrado, un microprocesador y/o un dispositivo de comunicación inalámbrica, tal como un teléfono móvil.

[0031] En un ejemplo de la divulgación, el descodificador de vídeo 30 se puede configurar para recibir un primer bloque codificado de datos de vídeo en una primera unidad de acceso de una primera vista, en el que el primer bloque codificado de datos de vídeo fue codificado usando predicción residual avanzada y predicción bidireccional, incluyendo la predicción bidireccional la predicción temporal para una primera dirección de predicción y la predicción inter vista para una segunda dirección de predicción. El descodificador de vídeo 30 se puede configurar además para determinar información de movimiento temporal para la primera dirección de predicción para el primer bloque codificado de datos de vídeo y determinar información de movimiento de disparidad para la segunda dirección de predicción para el primer bloque codificado de datos de vídeo. El descodificador de vídeo 30 se puede configurar además para identificar bloques de referencia para la segunda dirección de predicción, diferente de la primera dirección de predicción, usando la información de movimiento temporal determinada para la primera dirección de predicción, en el que los bloques de referencia están en una unidad de acceso diferente de la primera unidad de acceso, y realizar una predicción residual avanzada en el primer bloque codificado de datos de vídeo usando los bloques de referencia identificados para la segunda dirección de predicción. De esta forma, la información de movimiento temporal para la primera dirección de predicción se reutiliza para la segunda dirección de predicción. Como tal, se deben realizar menos accesos a memoria de la información de movimiento temporal, ya que no es necesario acceder a la información de movimiento temporal del bloque identificado por el vector de movimiento del primer bloque codificado correspondiente a la segunda dirección de predicción, por tanto permite una descodificación de vídeo más rápida. Además, el número total de bloques de referencia usados al realizar ARP se puede reducir de 6 a 5, lo que resulta en una menor complejidad computacional en términos de interpolación usando funcionamientos de multiplicación y suma. Asimismo, cuando se realiza la interpredicción bidireccional, el codificador de vídeo 20 se puede configurar para reutilizar la información de movimiento temporal para una primera dirección de predicción cuando se codifica la segunda dirección de predicción.

[0032] El codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con una norma de codificación de vídeo, tal como la norma de codificación de vídeo de alta eficacia (HEVC), actualmente en fase de desarrollo, y se pueden ajustar al modelo de prueba de la HEVC (HM). De forma alternativa, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo a otras normas privadas o industriales, tales como la

norma ITU-T H.264, de forma alternativa denominada MPEG-4, Parte 10, codificación de vídeo avanzada (AVC), o ampliaciones de dichas normas, como la ampliación MVC de ITU-T H.264/AVC. El último borrador conjunto de MVC se describe en "Advanced video coding for generic audiovisual services", Recomendación ITU-T H.264, marzo de 2010. En particular, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar de acuerdo con la norma de codificación 3D y/o multivista, incluyendo una extensión 3D de la norma HEVC (por ejemplo, 3D-HEVC).

[0033] Un borrador reciente de la norma de HEVC, denominado "HEVC Working Draft 10" o "WD10", se describe en el documento JCTVC-L1003v34, de Bross et al., titulado "High efficiency video coding (HEVC) text specification draft 10 (for FDIS & Last Call)", Equipo de Colaboración Conjunta sobre Codificación de Vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 12ª Reunión: Ginebra, Suiza, del 14 al 23 de enero de 2013, que, a partir del 5 de enero de 2015, se puede descargar desde: http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v34.zip.

[0034] Otro borrador de la norma HEVC, se denomina en el presente documento "WD10 revisions", descrito por Bross et al., "Editors' proposed corrections to HEVC version 1", Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 13ª reunión, Incheon, Corea, abril de 2013, que, desde 5 de enero de 2015, está disponible en: http://phenix.int-evry.fr/jct/doc_end_user/documents/13_Incheon/wg11/JCTVC-M0432-v3.zip. El JCT-3V también está desarrollando la extensión multivista de HEVC, concretamente, la MV-HEVC.

[0035] Actualmente, un Equipo de Colaboración Conjunta en Codificación de vídeo 3D (JCT-3C) de VCEG y MPEG está elaborando una norma de 3DV basada en la HEVC, para la cual una parte de los trabajos de normalización incluye la normalización del códec de vídeo multivista basado en la HEVC (MV-HEVC) y otra parte, la codificación de vídeo 3D basada en la HEVC (3D-HEVC). Para MV-HEVC, se debería garantizar que solo existan cambios sintácticos de alto nivel (HLS), de modo que ningún módulo en el nivel de unidad de codificación/unidad de predicción en HEVC necesite ser rediseñado y se pueda reutilizar por completo para MV-HEVC. Para la 3D-HEVC, se pueden incluir y dotar de soporte nuevas herramientas de codificación, incluyendo aquellas a nivel de unidad de codificación/unidad de predicción, tanto para las vistas de textura como de profundidad.

[0036] Una versión del programa informático 3D-HTM para la 3D-HEVC se puede descargar desde el siguiente enlace: [3D-HTM versión 9.01r1]: https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-9.0r1/. Una versión de la descripción de programa informático de referencia se describe en Li Zhang, Gerhard Tech, Krzysztof Wegner y Sehoon Yea, "Test Model 6 of 3D-HEVC and MV-HEVC", JCT3V-F1005, Equipo de colaboración conjunta en el desarrollo de la extensión de codificación de vídeo 3D del UIT-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 6ª Reunión: Ginebra, CH, noviembre de 2013 (JCT3V-F1005). JCT3V-F1005 se puede descargar desde http://phenix.it-sudparis.eu/jct2/doc_end_user/current_document.php?id=1636.

[0037] Un borrador de trabajo de 3D-HEVC se describe en Gerhard Tech, Krzysztof Wegner, Ying Chen y Sehoon Yea, "3D-HEVC Test Model 2", JCT3V-F1001, 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, 6ª Reunión. Ginebra, CH, noviembre de 2013 (JCT3V-F1-001). JCT3V-F1001 está disponible en http://phenix.it-sudparis.eu/jct2/doc_end_user/current_document.php?id=1361. La más reciente descripción del programa informático (número de documento: E1005) está disponible en: http://phenix.it-sudparis.eu/jct2/doc_end_user/current_document.php?id=1360.

[0038] Se puede descargar una versión más reciente del programa informático 3D-HTM para 3D-HEVC desde el siguiente enlace: [3D-HTM versión 12.0]: https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-12.0/. El correspondiente borrador de trabajo de 3D-HEVC (número de documento: 11001) está disponible en: http://phenix.int-evry.fr/jct3v/doc_end_user/current_document.php?id=2299. La más reciente descripción del programa informático (número de documento: 11005) está disponible en: http://phenix.int-evry.fr/jct3v/doc_end_user/current_document.php?id=2301.

[0039] Inicialmente, se analizarán las técnicas de codificación de ejemplo de HEVC. Los trabajos de normalización de HEVC se basaron en un modelo en evolución de un dispositivo de codificación de vídeo denominado modelo de prueba de HEVC (HM). El HM supone diversas capacidades adicionales de los dispositivos de codificación de vídeo relativos a los dispositivos existentes de acuerdo con, por ejemplo, ITU-T H.264/AVC. Por ejemplo, mientras que la norma H.264 proporciona nueve modos de codificación intrapredicción, el HM puede proporcionar hasta treinta y tres modos de codificación intrapredicción angulares, más modos DC y planares.

[0040] En HEVC y otras memorias descriptivas de codificación de vídeo, una secuencia de vídeo incluye típicamente una serie de imágenes. Las imágenes también se pueden denominar "tramas". Una imagen puede incluir tres matrices de muestras, indicadas como S_L , S_{Cb} y S_{Cr} . S_L es una matriz bidimensional (es decir, un bloque) de muestras de luma. S_{Cb} es una matriz bidimensional de muestras de croma Cb. S_{Cr} es una matriz bidimensional de muestras de croma Cr. Las muestras de croma también se pueden denominar en el presente documento muestras de "croma". En otros casos, una imagen puede ser monocromática y puede incluir solo una matriz de muestras de luma.

[0041] Para generar una representación codificada de una imagen, el codificador de vídeo 20 puede generar un conjunto de unidades de árbol de codificación (CTU). Cada una de las CTU puede comprender un bloque en árbol de codificación de muestras de luma, dos bloques en árbol de codificación correspondientes de muestras de croma y unas estructuras sintácticas usadas para codificar las muestras de los bloques en árbol de codificación. En imágenes monocromáticas o imágenes que tengan tres planos de color separados, una CTU puede comprender un solo bloque en árbol de codificación y unas estructuras sintácticas usadas para codificar las muestras del bloque en árbol de codificación. Un bloque en árbol de codificación puede ser un bloque de muestras NxN. Una CTU también se puede denominar "bloque en árbol" o "unidad de codificación de máximo tamaño" (LCU). Las CTU de la HEVC pueden ser análogas en términos generales a los macrobloques de otros estándares, tales como H.264/AVC. Sin embargo, una CTU no está necesariamente limitada a un tamaño en particular y puede incluir una o más unidades de codificación (CU). Un fragmento puede incluir un número entero de CTU ordenadas consecutivamente en un orden de escaneo de barrido.

[0042] Para generar una CTU codificada, el codificador de vídeo 20 puede llevar a cabo de forma recursiva una partición de árbol cuádruple en los bloques de árbol de codificación de una CTU para dividir los bloques de árbol de codificación en bloques de codificación, de ahí el nombre "unidades de árbol de codificación." Un bloque de codificación es un bloque NxN de muestras. Una unidad de codificación (CU) puede comprender un bloque de codificación de muestras de luma y dos bloques de codificación correspondientes de muestras de croma de una imagen que tiene una matriz de muestras de luma, una matriz de muestras Cb y una matriz de muestras Cr y estructuras sintácticas usadas para codificar las muestras de los bloques de codificación. En imágenes monocromáticas o imágenes que tengan tres planos de color separados, una CU puede comprender un único bloque de codificación y estructuras sintácticas usadas para codificar las muestras del bloque de codificación.

[0043] El codificador de vídeo 20 puede dividir un bloque de codificación de una CU en uno o más bloques de predicción. Un bloque de predicción es un bloque rectangular (es decir, cuadrado o no cuadrado) de muestras en las cuales se aplica la misma predicción. Una unidad de predicción (PU) de una CU puede comprender un bloque de predicción de muestras de luma, dos bloques de predicción correspondientes de muestras de croma y estructuras sintácticas usadas para predecir los bloques de predicción. En imágenes monocromáticas o imágenes que tengan tres planos de color separados, una PU puede comprender un único bloque de predicción y estructuras sintácticas usadas para predecir el bloque de predicción. El codificador de vídeo 20 puede generar bloques predictivos de luma, Cb y Cr para bloques de predicción de luma, Cb y Cr de cada PU de la CU.

[0044] El codificador de vídeo 20 puede usar intrapredicción o interpredicción para generar los bloques predictivos para una PU. Si el codificador de vídeo 20 usa intrapredicción para generar los bloques predictivos de una PU, el codificador de vídeo 20 puede generar los bloques predictivos de la PU en base a muestras descodificadas de la imagen asociada con la PU. En algunas versiones de HEVC, para el componente de luma de cada PU, se utiliza un procedimiento de intrapredicción con 33 modos de predicción angular (indizados de 2 a 34), el modo DC (indizado con 1) y el modo Planar (indizado con 0).

[0045] Si el codificador de vídeo 20 usa interpredicción para generar los bloques predictivos de una PU, el codificador de vídeo 20 puede generar los bloques predictivos de la PU en base a muestras descodificadas de una o más imágenes distintas a la imagen asociada con la PU. La interpredicción puede ser una interpredicción unidireccional (es decir, unipredicción o predicción unipredictiva) o una interpredicción bidireccional (es decir, bipredicción o predicción bipredictiva). Para realizar la unipredicción o la bipredicción, el codificador de vídeo 20 puede generar una primera lista de imágenes de referencia (RefPicList0) y una segunda lista de imágenes de referencia (RefPicList1) para un fragmento actual. Cada una de las listas de imágenes de referencia puede incluir una o más imágenes de referencia. Cuando se usa unipredicción, el codificador de vídeo 20 puede buscar las imágenes de referencia en RefPicList0, RefPicList1 o en ambas para determinar una localización de referencia dentro de una imagen de referencia. Además, cuando se usa unipredicción, el codificador de vídeo 20 puede generar, en base, al menos en parte, a muestras correspondientes a la localización de referencia, los bloques predictivos de muestras para la PU. Además, cuando se usa la unipredicción, el codificador de vídeo 20 puede generar un único vector de movimiento que indique un desplazamiento espacial entre un bloque de predicción de la PU y la localización de referencia. Para indicar el desplazamiento espacial entre un bloque de predicción de la PU y la localización de referencia, un vector de movimiento puede incluir un componente horizontal que especifique un desplazamiento horizontal entre el bloque de predicción de la PU y la localización de referencia y puede incluir un componente vertical que especifique un desplazamiento vertical entre el bloque de predicción de la PU y la localización de referencia.

[0046] Cuando se usa la bipredicción para codificar una PU, el codificador de vídeo 20 puede determinar una primera localización de referencia en una imagen de referencia en RefPicList0 y una segunda localización de referencia en una imagen de referencia en RefPicList1. El codificador de vídeo 20 puede generar entonces, basándose, al menos en parte, en muestras correspondientes a las primera y segunda localizaciones de referencia, los bloques predictivos para la PU. Además, cuando se usa la bipredicción para codificar la PU, el codificador de vídeo 20 puede generar un primer vector de movimiento que indica un desplazamiento espacial entre un bloque de muestra de la PU y la primera localización de referencia y un segundo vector de movimiento que indica un desplazamiento espacial entre el bloque de predicción de la PU y la segunda localización de referencia.

[0047] Típicamente, una estructura de lista de imágenes de referencia para la primera o la segunda lista de imágenes de referencia (por ejemplo, RefPicList0 o RefPicList1) de una imagen B incluye dos etapas: inicialización de lista de imágenes de referencia y reordenamiento (modificación) de la lista de imágenes de referencia. La inicialización de lista de imágenes de referencia es un mecanismo explícito que introduce las imágenes de referencia en la memoria de imágenes de referencia (también conocida como memoria intermedia de imágenes descodificadas) como una lista en base al orden de los valores POC (recuento de orden de imágenes, alineado con el orden de visualización de una imagen). El mecanismo de reordenación de lista de imágenes de referencia puede modificar la posición de una imagen que se introdujo en la lista durante la inicialización de lista de imágenes de referencia en cualquier posición nueva, o introducir cualquier imagen de referencia en la memoria de imágenes de referencia en cualquier posición, incluso si la imagen no pertenece a la lista inicializada. Algunas imágenes, después del reordenamiento (modificación) de la lista de imágenes de referencia, se pueden colocar en posiciones muy lejanas en la lista. Sin embargo, si la posición de una imagen supera un número de imágenes de referencia activas de la lista, la imagen no se considera una entrada de la lista final de imágenes de referencia. El número de imágenes de referencia activas se puede señalar en la cabecera de fragmento para cada lista.

[0048] Después de que se construya las listas de imágenes de referencia (concretamente, RefPicList0 y RefPicList1, si disponible), se puede usar un índice de referencia a una lista de imágenes de referencia para identificar cualquier imagen de referencia incluida en la lista de imágenes de referencia.

[0049] Después de que el codificador de vídeo 20 genera luma predictivo, bloques Cb y Cr para una o más PU de una CU, el codificador de vídeo 20 puede generar un bloque residual de luma para la CU. Cada muestra del bloque residual de luma de la CU indica una diferencia entre una muestra de luma en uno de los bloques predictivos de luma de la CU y una muestra correspondiente en el bloque de codificación de luma original de la CU. Además, el codificador de vídeo 20 puede generar un bloque residual de Cb para la CU. Cada muestra del bloque residual de Cb de la CU puede indicar una diferencia entre una muestra de Cb de uno de los bloques predictivos de Cb de la CU y una muestra correspondiente del bloque de codificación de Cb original de la CU. El codificador de vídeo 20 también puede generar un bloque residual de Cr para la CU. Cada muestra del bloque residual de Cr de la CU puede indicar una diferencia entre una muestra de Cr en uno de los bloques predictivos de Cr de la CU y una muestra correspondiente en el bloque de codificación de Cr original de la CU.

[0050] Además, el codificador de vídeo 20 puede usar la partición en árbol cuádruple para descomponer los bloques residuales de luma, de Cb y de Cr de una CU en uno o más bloques de transformada de luma, de Cb y de Cr. Un bloque de transformada es un bloque rectangular (por ejemplo, cuadrado o no cuadrado) de muestras a las que se aplica la misma transformada. Una unidad de transformada (TU) de una CU puede comprender un bloque de transformada de muestras de luma, dos bloques correspondientes de transformada de muestras de croma y estructuras sintácticas usadas para transformar las muestras de bloques de transformada. Por tanto, cada TU de una CU puede estar asociada con un bloque de transformada de luma, un bloque de transformada de Cb y un bloque de transformada de Cr. El bloque de transformada de luma asociado a la TU puede ser un subbloque del bloque residual de luma de la CU. El bloque de transformada de Cb puede ser un subbloque del bloque residual de Cb de la CU. El bloque de transformada de Cr puede ser un subbloque del bloque residual de Cr de la CU. En imágenes monocromáticas o imágenes que tengan tres planos de color separados, una TU puede comprender un único bloque de transformada y estructuras sintácticas usadas para transformar las muestras del bloque de transformada.

[0051] El codificador de vídeo 20 puede aplicar una o más transformadas a un bloque de transformada de luma de una TU con el fin de generar un bloque de coeficientes luma para la TU. Un bloque de coeficientes puede ser una matriz bidimensional de coeficientes de transformada. Un coeficiente de transformada puede ser una cantidad escalar. El codificador de vídeo 20 puede aplicar una o más transformadas a un bloque de transformada de Cb de una TU para generar un bloque de coeficientes de Cb para la TU. El codificador de vídeo 20 puede aplicar una o más transformadas a un bloque de transformada de Cr de una TU para generar un bloque de coeficientes de Cr para la TU.

[0052] Después de generar un bloque de coeficientes (por ejemplo, un bloque de coeficientes luma, un bloque de coeficientes Cb o un bloque de coeficientes Cr), el codificador de vídeo 20 puede cuantificar el bloque de coeficientes. La cuantificación se refiere, en general, a un proceso en el que coeficientes de transformada se cuantifican para reducir, posiblemente, la cantidad de datos usados para representar los coeficientes de transformada, proporcionando más compresión. Después de que el codificador de vídeo 20 cuantifique un bloque de coeficientes, el codificador de vídeo 20 puede codificar por entropía elementos sintácticos que indiquen los coeficientes de transformada cuantificados. Por ejemplo, el codificador de vídeo 20 puede realizar una codificación aritmética binaria adaptativa al contexto (CABAC) en los elementos sintácticos que indiquen los coeficientes de transformada cuantificados.

[0053] El codificador de vídeo 20 puede emitir un flujo de bits que incluya una secuencia de bits que forme una representación de imágenes codificadas y datos asociados. El flujo de bits puede comprender una secuencia de unidades de capa de abstracción de red (NAL). Una unidad NAL es una estructura sintáctica que contiene una indicación del tipo de datos en la unidad NAL y octetos que contienen esos datos en forma de una carga útil de secuencia de octetos sin procesar (RBSP) entremezclados como sea necesario con bits de prevención de emulación. Cada una de las unidades NAL incluye una cabecera de unidad NAL y encapsula una RBSP. La cabecera de unidad

NAL puede incluir un elemento sintáctico que indique un código de tipo de unidad NAL. El código de tipo de unidad NAL especificado por la cabecera de unidad NAL de una unidad NAL indica el tipo de la unidad NAL. Una RBSP puede ser una estructura sintáctica que contenga un número entero de bytes que esté encapsulado dentro de una unidad NAL. En algunos casos, una RBSP incluye cero bits.

[0054] Diferentes tipos de unidades de NAL pueden encapsular diferentes tipos de RBSP. Por ejemplo, un primer tipo de unidad NAL puede encapsular una RBSP para un conjunto de parámetros de imagen (PPS), un segundo tipo de unidad NAL puede encapsular una RBSP para un fragmento codificado, un tercer tipo de unidad NAL puede encapsular una RBSP para SEI, etc. Las unidades NAL que encapsulan las RBSP para datos de codificación de vídeo (a diferencia de las RBSP para conjuntos de parámetros y mensajes SEI) se pueden denominar unidades NAL de capa de codificación de vídeo (VCL).

[0055] El descodificador de vídeo 30 puede recibir un flujo de bits generado por el codificador de vídeo 20. Además, el descodificador de vídeo 30 puede analizar el flujo de bits para obtener elementos sintácticos a partir del flujo de bits. El descodificador de vídeo 30 puede reconstruir las imágenes de los datos de vídeo en base a, al menos en parte, los elementos sintácticos obtenidos a partir del flujo de bits. El procedimiento para reconstruir los datos de vídeo puede ser, en general, recíproco al procedimiento realizado por el codificador de vídeo 20. Por ejemplo, el descodificador de vídeo 30 puede usar vectores de movimiento de las PU para determinar bloques predictivos para las PU de una CU actual. Además, el descodificador de vídeo 30 puede cuantificar de forma inversa bloques de coeficientes asociados a las TU de la CU actual. El descodificador de vídeo 30 puede realizar transformadas inversas en los bloques de coeficientes para reconstruir los bloques de transformada asociados a las TU de la CU actual. El descodificador de vídeo 30 puede reconstruir los bloques de codificación de la CU actual añadiendo las muestras de los bloques predictivos para las PU de la CU actual a las muestras correspondientes de los bloques de transformada de las TU de la CU actual. Reconstruyendo los bloques de codificación para cada CU de una imagen, el descodificador de vídeo 30 puede reconstruir la imagen.

[0056] En algunos ejemplos, el codificador de vídeo 20 puede señalar la información de movimiento de una PU usando el modo de fusión o modo de predicción de movimiento avanzada (AMVP). En otras palabras, en HEVC, existen dos modos para la predicción de parámetros de movimiento, uno siendo el modo de fusión y el otro siendo AMVP. La predicción de movimiento puede comprender la determinación de la información de movimiento de una unidad de vídeo (por ejemplo, una PU) en base a la información de movimiento de una o más otras unidades de vídeo. La información de movimiento de una PU puede incluir uno o más vectores de movimiento de la PU y uno o más índices de referencia de la PU.

[0057] Cuando el codificador de vídeo 20 señala la información de movimiento de una PU actual usando el modo de fusión, el codificador de vídeo 20 genera una lista de candidatos de fusión. En otras palabras, el codificador de vídeo 20 puede realizar un proceso de construcción de una lista de predictores de vectores de movimiento. La lista de candidatos de fusión incluye un conjunto de candidatos de fusión que indican la información de movimiento de las PU que se encuentran próximas espacial o temporalmente a la PU actual. Es decir, en el modo de fusión, se construye una lista de candidatos de parámetros de movimiento (por ejemplo, índices de referencia, vectores de movimiento, etc.) donde el candidato puede ser de bloques contiguos espaciales y temporales. En algunos ejemplos, los candidatos también pueden incluir un candidato generado artificialmente.

[0058] Además, el codificador de vídeo 20 puede seleccionar un candidato de fusión de la lista de candidatos de fusión y puede usar la información de movimiento indicada por el candidato de fusión seleccionado como la información de movimiento de la PU actual. El codificador de vídeo 20 puede señalar la posición en la lista de candidatos de fusión del candidato de fusión seleccionado. Por ejemplo, el codificador de vídeo 20 puede señalar los parámetros del vector de movimiento seleccionados transmitiendo un índice a la lista de candidatos. El descodificador de vídeo 30 puede obtener, a partir del flujo de bits, el índice en la lista de candidatos (es decir, un índice de lista de candidatos). El descodificador de vídeo 30 puede generar la misma lista de candidatos de fusión y puede determinar, en base a la indicación de la posición del candidato de fusión seleccionado, el candidato de fusión seleccionado. El descodificador de vídeo 30 puede usar a continuación la información de movimiento del candidato de fusión seleccionado para generar bloques predictivos para la PU actual. Es decir, el descodificador de vídeo 30 puede determinar, en base al menos en parte al índice de la lista de candidatos, un candidato seleccionado en la lista de candidatos, en el que el candidato seleccionado especifica el vector de movimiento para la PU actual. De esta forma, en el lado del descodificador, una vez descodificado el índice, todos los parámetros de movimiento del bloque correspondiente donde apunta el índice pueden ser heredados por la PU actual.

[0059] El modo de omisión es similar al modo de fusión. En el modo de omisión, el codificador de vídeo 20 y el descodificador de vídeo 30 generan y usan una lista de candidatos de fusión de la misma manera que el codificador de vídeo 20 y el descodificador de vídeo 30 usan la lista de candidatos de fusión en el modo de fusión. Sin embargo, cuando el codificador de vídeo 20 señala la información de movimiento de una PU actual usando el modo de omisión, el codificador de vídeo 20 no señala ningún dato residual para la PU actual. En consecuencia, el descodificador de vídeo 30 puede determinar, sin usar datos residuales, un bloque predictivo para la PU en base a un bloque de referencia indicado por la información de movimiento de un candidato seleccionado en la lista de candidatos de fusión.

[0060] El modo AMVP es similar al modo de fusión en cuanto a que el codificador de vídeo 20 puede generar una lista de candidatos y puede seleccionar un candidato de la lista de candidatos. Sin embargo, cuando el codificador de vídeo 20 señala la información de movimiento RefPicListX de una PU actual usando el modo AMVP, el codificador de vídeo 20 puede señalar una diferencia de vector de movimiento (MVD) RefPicListX para la PU actual y un índice de referencia RefPicListX para la PU actual además de señalar un indicador RefPicListX MVP para la PU actual. El indicador RefPicListX MVP para la PU actual puede indicar la posición de un candidato AMVP seleccionado en la lista de candidatos AMVP. La RefPicListX MVD para la PU actual puede indicar una diferencia entre un vector de movimiento RefPicListX de la PU actual y un vector de movimiento del candidato AMVP seleccionado. De esta forma, el codificador de vídeo 20 puede señalar la información de movimiento RefPicListX de la PU actual señalizando un indicador de predictor de vector de movimiento (MVP) RefPicListX, un valor de índice de referencia RefPicListX y una RefPicListX MVD. En otras palabras, los datos en el flujo de bits que representan el vector de movimiento para la PU actual pueden incluir datos que representan un índice de referencia, un índice a una lista de candidatos y una MVD.

[0061] Además, cuando la información de movimiento de una PU actual se señala usando el modo AMVP, el descodificador de vídeo 30 puede obtener, a partir del flujo de bits, una MVD para una PU actual y un indicador MVP. El descodificador de vídeo 30 puede generar la misma lista de candidatos AMVP y puede determinar, en base al indicador MVP, el candidato AMVP seleccionado. El descodificador de vídeo 30 puede recuperar un vector de movimiento de la PU actual añadiendo una MVD al vector de movimiento del candidato AMVP seleccionado. Es decir, el descodificador de vídeo 30 puede determinar, en base a un vector de movimiento indicado por el candidato AMVP seleccionado y la MVD, el vector de movimiento de la PU actual. El descodificador de vídeo 30 puede a continuación usar los vector o vectores de movimiento recuperados de la PU actual para generar bloques predictivos para la PU actual.

[0062] Cuando el descodificador de vídeo 30 genera una lista de candidatos AMVP para una PU actual, el descodificador de vídeo 30 puede derivar uno o más candidatos AMVP en base a la información de movimiento de las PU (es decir, PU contiguas espacialmente) que cubren localizaciones contiguas espacialmente a la PU actual. Una PU puede cubrir una localización cuando un bloque de predicción de la PU incluye la localización.

[0063] Un candidato en una lista de candidatos de fusión o una lista de candidatos de AMVP que se basa en la información de movimiento de una PU que se encuentra temporalmente próxima a una PU actual (es decir, una PU que está en una instancia de tiempo diferente a la PU actual) se puede denominar un TMVP. Es decir, un TMVP se puede usar para mejorar la eficacia de codificación de HEVC y, a diferencia de otras herramientas de codificación, TMVP puede necesitar acceder al vector de movimiento de una trama en una memoria intermedia de imágenes descodificadas, más específicamente en una lista de imágenes de referencia.

[0064] El uso de TMVP se puede habilitar o deshabilitar sobre una base de CVS-por-CVS (secuencia de vídeo codificada), base de fragmento por fragmento, o sobre otra base. Un elemento de sintaxis (por ejemplo, `sps_temporal_mvp_enable_flag`) en un SPS puede indicar si el uso de TMVP está habilitado para una CVS. Además, cuando el uso de TMVP está habilitado para una CVS, el uso de TMVP se puede habilitar o deshabilitar para fragmentos particulares dentro de la CVS. Por ejemplo, un elemento de sintaxis (por ejemplo, `slice_temporal_mvp_enable_flag`) en un encabezado de fragmento puede indicar si el uso de TMVP está habilitado para un fragmento. Por tanto, en un fragmento inter-predicho, cuando el TMVP está habilitado para una CVS completa (por ejemplo, `sps_temporal_mvp_enable_flag` en un SPS se establece en 1), se señala `slice_temporal_mvp_enable_flag` en el encabezado del fragmento para indicar si el TMVP está habilitado para el fragmento actual.

[0065] Para determinar un TMVP, un codificador de vídeo puede identificar en primer lugar una imagen de referencia que incluye una PU que está coubicada con la PU actual. En otras palabras, el codificador de vídeo puede identificar una imagen coubicada. Si el fragmento actual de la imagen actual es un fragmento B (es decir, un fragmento que puede incluir PU inter-predichas bidireccionalmente), el codificador de vídeo 20 puede señalar, en un encabezado de fragmento, un elemento de sintaxis (por ejemplo, `collocated_from_I0_flag`) que indica si la imagen coubicada es de RefPicList0 o RefPicList1. En otras palabras, cuando el uso de TMVP está habilitado para un fragmento actual, y el fragmento actual es un fragmento B (por ejemplo, un fragmento que puede incluir PU inter-predichas bidireccionalmente), el codificador de vídeo 20 puede señalar un elemento de sintaxis (por ejemplo, `collocated_from_I0_flag`) en un encabezado de fragmento para indicar si la imagen coubicada está en RefPicList0 o RefPicList1. En otras palabras, para obtener un TMVP, primero se debe identificar una imagen coubicada. Si la imagen actual es un fragmento B, se señala un `collocated_from_I0_flag` en el encabezado de fragmento para indicar si la imagen coubicada es de RefPicList0 o RefPicList1.

[0066] Después de que el descodificador de vídeo 30 identifica la lista de imágenes de referencia que incluye la imagen coubicada, el descodificador de vídeo 30 puede usar otro elemento de sintaxis (por ejemplo, `collocated_ref_idx`), que se puede señalar en un encabezado de fragmento, para identificar una imagen (es decir, la imagen coubicada) en la lista de imágenes de referencia identificada. Es decir, después de identificar una lista de imágenes de referencia, `collocated_ref_idx` señalado en el encabezado de fragmento, se usa para identificar la imagen en la lista de imágenes de referencia.

[0067] El codificador de vídeo puede identificar una PU coubicada verificando la imagen coubicada. El TMVP puede indicar la información de movimiento de una PU inferior derecha de la CU que contiene la PU coubicada, o la información de movimiento de la PU inferior derecha dentro de las PU centrales de la CU que contiene esta PU. Por tanto, se usa el movimiento de la PU inferior derecha de la CU que contiene esta PU, o el movimiento de la PU inferior derecha dentro de las PU centrales de la CU que contiene esta PU. La PU inferior derecha de la CU que contiene la PU coubicada puede ser una PU que cubre una localización inmediatamente debajo y a la derecha de una muestra inferior derecha de un bloque de predicción de la PU. En otras palabras, el TMVP puede indicar la información de movimiento de una PU que está en la imagen de referencia y que cubre una localización que se coubica con una esquina inferior derecha de la PU actual, o el TMVP puede indicar la información de movimiento de una PU que está en la imagen de referencia y que cubre una localización que está coubicada con un centro de la PU actual.

[0068] Cuando los vectores de movimiento identificados por el proceso anterior (es decir, los vectores de movimiento de un TMVP) se usan para generar un candidato de movimiento para el modo de fusión o el modo AMVP, el codificador de vídeo puede escalar los vectores de movimiento en base a la localización temporal (reflejada por el valor POC). Por ejemplo, un codificador de vídeo puede incrementar la magnitud de un vector de movimiento en cantidades mayores cuando una diferencia entre los valores POC de una imagen actual y una imagen de referencia es mayor e incrementar la magnitud del vector de movimiento en cantidades menores cuando una diferencia entre los valores POC de la imagen actual y la imagen de referencia son menores.

[0069] El índice de referencia objetivo de todas las posibles listas de imágenes de referencia para el candidato de fusión temporal derivado de un TMVP se puede establecer siempre en 0. Sin embargo, para AMVP, el índice de referencia objetivo de todas las imágenes de referencia posibles se establece igual al índice de referencia descodificado. En otras palabras, el índice de referencia objetivo de todas las listas de imágenes de referencia posibles para el candidato de fusión temporal obtenido del TMVP se establece siempre en 0, mientras que, para el AMVP, se establece igual al índice de referencia descodificado. En la HEVC, el SPS puede incluir un indicador (por ejemplo, `sps_temporal_mvp_enable_flag`) y el encabezado de fragmento puede incluir un indicador (por ejemplo, `pic_temporal_mvp_enable_flag`) cuando `sps_temporal_mvp_enable_flag` es igual a 1. Cuando tanto `pic_temporal_mvp_enable_flag` como una id temporal son iguales a 0 para una imagen particular, ningún vector de movimiento de imágenes antes de esa imagen particular en orden de descodificación se usa como un TMVP para descodificar la imagen particular o una imagen después de la imagen particular en orden de descodificación.

[0070] En algunos ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 (FIG. 1) pueden emplear técnicas para la codificación de vídeo multivista y/o 3D, por ejemplo, la codificación de datos de vídeo que incluyen dos o más vistas. En tales ejemplos, el codificador de vídeo 20 puede codificar un flujo de bits que incluye datos de vídeo codificados para dos o más vistas, y el descodificador de vídeo 30 puede descodificar los datos de vídeo codificados para proporcionar las dos o más vistas, por ejemplo, al dispositivo de visualización 32. En algunos ejemplos, el descodificador de vídeo 30 puede proporcionar las múltiples vistas de datos de vídeo para permitir al dispositivo de visualización 32 exhibir vídeo 3D. En algunos ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden ser conformes a una extensión 3D-HEVC de la norma HEVC, por ejemplo, en la cual se usan procesos de codificación multivista y de codificación multivista con profundidad. La codificación de vídeo multivista y/o 3D, puede implicar la codificación de dos o más vistas de textura, y/o vistas que incluyan componentes de textura y de profundidad. En algunos ejemplos, los datos de vídeo codificados por el codificador de vídeo 20 y descodificados por el descodificador de vídeo 30 incluyen dos o más imágenes en cualquier instancia temporal dada, es decir, dentro de una "unidad de acceso", o datos de los cuales pueden ser obtenidas dos o más imágenes en cualquier instancia temporal dada.

[0071] En algunos ejemplos, un dispositivo, por ejemplo, la fuente de vídeo 18, puede generar las dos o más imágenes, por ejemplo, usando dos o más cámaras espacialmente desplazadas, u otros dispositivos de captura de vídeo, para capturar una escena común. Dos imágenes de la misma escena capturadas simultáneamente, o casi simultáneamente, desde posiciones horizontales levemente distintas, pueden ser usadas para producir un efecto tridimensional. En algunos ejemplos, la fuente de vídeo 18 (u otro componente del dispositivo de origen 12) puede usar información de profundidad o información de disparidad para generar una segunda imagen (u otra adicional) de una segunda vista (u otra adicional) en una instancia temporal dada, a partir de una primera imagen de una primera vista en la instancia temporal dada. En este caso, una vista dentro de una unidad de acceso puede incluir un componente de textura correspondiente a una primera vista y un componente de profundidad que puede ser usado, con el componente de textura, para generar una segunda vista. La información de profundidad o disparidad puede ser determinada por un dispositivo de captura de vídeo que captura la primera vista, por ejemplo, en base a parámetros de cámara u otra información conocida, con respecto a la configuración del dispositivo de captura de vídeo y a la captura de los datos de vídeo para la primera vista. La información de profundidad o disparidad puede ser calculada, adicionalmente o de forma alternativa, por la fuente de vídeo 18 u otro componente del dispositivo de origen 12, a partir de parámetros de cámara y/o datos de vídeo en la primera vista.

[0072] Para presentar vídeo 3D, el dispositivo de visualización 32 puede mostrar, simultáneamente, o casi simultáneamente, dos imágenes asociadas a vistas diferentes de una escena común, que fueron capturadas simultáneamente, o casi simultáneamente. En algunos ejemplos, un usuario del dispositivo objetivo 14 puede usar gafas activas para obturar, rápidamente y de forma alternativa, las lentes izquierda y derecha, y el dispositivo de

visualización 32 puede conmutar rápidamente entre una vista izquierda y una vista derecha, en sincronización con las gafas activas. En otros ejemplos, el dispositivo de visualización 32 puede mostrar las dos vistas simultáneamente, y el usuario puede usar gafas pasivas, por ejemplo, con lentes polarizadas, que filtran las vistas para hacer que las vistas adecuadas pasen a través hasta los ojos del usuario. En otros ejemplos, el dispositivo de visualización 32 puede comprender un visor auto-estereoscópico, que no requiere gafas para que el usuario perciba el efecto 3D.

[0073] La codificación de vídeo multivista se refiere a la manera en la que se codifica una pluralidad de vistas. En el caso de la codificación de vídeo en 3D, la pluralidad de vistas puede, por ejemplo, corresponder a una vista de ojo izquierdo y una vista de ojo derecho. Cada vista de la pluralidad de vistas incluye una pluralidad de imágenes. La percepción del espectador de una escena 3D se debe a la disparidad horizontal entre los objetos en las imágenes de las diferentes vistas.

[0074] Un vector de disparidad (DV) para un bloque actual de una imagen actual es un vector que apunta a un bloque correspondiente en una imagen correspondiente que está en una vista diferente a la imagen actual. Por tanto, usando un DV, un codificador de vídeo puede localizar, en una imagen correspondiente, el bloque que corresponde a un bloque actual de una imagen actual. En este caso, la imagen correspondiente es una imagen que es de la misma instancia temporal que la imagen actual pero que está en una vista diferente. El bloque correspondiente en la imagen correspondiente y el bloque actual en la imagen actual pueden incluir contenido de vídeo similar; sin embargo, existe al menos una disparidad horizontal entre la localización del bloque actual en la imagen actual y la localización del bloque correspondiente en la imagen correspondiente. El DV del bloque actual proporciona una medida de esta disparidad horizontal entre el bloque en la imagen correspondiente y el bloque actual en la imagen actual.

[0075] En algunos casos, también puede existir disparidad vertical entre la localización del bloque dentro de la imagen correspondiente y la localización del bloque actual dentro de la imagen actual. El DV del bloque actual también puede proporcionar una medida de esta disparidad vertical entre el bloque en la imagen correspondiente y el bloque actual en la imagen actual. Un DV contiene dos componentes (un componente x y un componente y), aunque en muchos casos el componente vertical será igual a cero. El tiempo en que se muestra la imagen actual de la vista actual y la imagen correspondiente de la vista diferente puede ser el mismo, es decir, la imagen actual y las imágenes correspondientes son imágenes de la misma instancia temporal.

[0076] En la codificación de vídeo, en general hay dos tipos de predicción, comúnmente denominados intrapredicción e interpredicción. En la intrapredicción, un codificador de vídeo predice un bloque de vídeo en una imagen en base a un bloque ya codificado en la misma imagen. En la interpredicción, un codificador de vídeo predice un bloque de vídeo en una imagen en base a un bloque ya codificado de una imagen diferente (es decir, una imagen de referencia). Una imagen de referencia, tal como se usa en esta divulgación, se refiere en general a cualquier imagen que contenga muestras que puedan ser usadas para la interpredicción en el proceso de decodificación de imágenes posteriores en orden de decodificación. Cuando se codifica un contenido multivista, por ejemplo de acuerdo con 3D-HEVC, con respecto a una imagen actual, las imágenes de referencia pueden ser de la misma instancia temporal pero en una vista diferente o pueden estar en la misma vista pero de una instancia temporal diferente. En el caso de la codificación multivista, por ejemplo, en la 3D-HEVC, la predicción entre imágenes puede incluir la predicción del bloque de vídeo actual, por ejemplo, el nodo de codificación actual de una CU, a partir de otro bloque de vídeo en una imagen temporalmente distinta, es decir, a partir de una unidad de acceso distinta a la de la imagen actual, así como la predicción a partir de una imagen distinta en la misma unidad de acceso que la imagen actual, pero asociada a una vista distinta a la de la imagen actual.

[0077] En el último caso de la interpredicción, se puede denominar codificación entrevista o predicción entrevista. La imagen de referencia que está en la misma unidad de acceso que la imagen actual, pero asociada con una vista diferente a la imagen actual, se puede denominar imagen de referencia entrevista. En la codificación multivista, la predicción de entrevista se realiza entre las imágenes capturadas en las diferentes vistas de la misma unidad de acceso (es decir, con la misma instancia de tiempo) para eliminar la correlación entre las vistas. Al codificar una imagen de una vista no de base, por ejemplo, una vista dependiente, una imagen de la misma unidad de acceso, pero de una vista diferente, por ejemplo, de una vista de referencia, tal como la vista de base, se puede añadir a una lista de imágenes de referencia. Una imagen de referencia entrevista se puede poner en cualquier posición de una lista de imágenes de referencia, como es el caso con cualquier imagen de referencia de interpredicción (por ejemplo, temporal o entrevista).

[0078] El bloque de la imagen de referencia usado para predecir el bloque de la imagen actual se identifica mediante un vector de movimiento. En la codificación multivista, hay al menos dos clases de vectores de movimiento. Un vector de movimiento temporal (TMV) es un vector de movimiento que apunta a un bloque en una imagen de referencia temporal que está en la misma vista que el bloque que está siendo codificado, (por ejemplo, el primer ejemplo de interpredicción como se describe anteriormente), pero en una instancia de tiempo o unidad de acceso diferente al bloque que está siendo codificado, y la interpredicción correspondiente se denomina predicción con compensación de movimiento (MCP). Otro tipo de vector de movimiento es un vector de movimiento de disparidad (DMV), que apunta a un bloque en una imagen en la misma imagen actual de unidad de acceso, pero de una vista distinta. Con un DMV, la correspondiente inter-predicción se denomina predicción con compensación de disparidad (DCP) o predicción de entrevista.

[0079] En la siguiente sección, se analizarán las técnicas de codificación multivista (por ejemplo, como en H.264/MVC) y multivista más profundidad (por ejemplo, como en 3D-HEVC). Inicialmente, se analizarán las técnicas de MVC. Como se indica anteriormente, MVC es una extensión de codificación multivista de ITU-T H.264/AVC. En MVC, los datos para una pluralidad de vistas se codifican en orden de primer tiempo, y por consiguiente, la disposición de orden de descodificación se denomina codificación de primer tiempo. En particular, se pueden codificar componentes de vista (es decir, imágenes) para cada una de la pluralidad de vistas en una instancia de tiempo común, y a continuación se puede codificar otro conjunto de componentes de vista para una instancia de tiempo diferente, y así sucesivamente. Una unidad de acceso puede incluir imágenes codificadas de todas las vistas para una instancia de tiempo de salida. Se debe comprender que el orden de descodificación de las unidades de acceso no es necesariamente idéntico al orden de salida (o de visualización).

[0080] Una orden de descodificación MVC típica (es decir, orden de flujo de bits) se muestra en la FIG. 2. La disposición del orden de descodificación se denomina codificación de tiempo en primer lugar. Se debe tener en cuenta que el orden de descodificación de las unidades de acceso puede no ser idéntico al orden de salida o de visualización. En la FIG. 2, S0 a S7 se refieren, cada uno, a diferentes vistas del vídeo multivista. Cada una entre T0 a T8 representa una instancia de tiempo de salida. Una unidad de acceso puede incluir las imágenes codificadas de todas las vistas para una instancia de tiempo de salida. Por ejemplo, una primera unidad de acceso puede incluir todas las vistas S0 a S7 para la instancia de tiempo T0, una segunda unidad de acceso puede incluir todas las vistas S0 a S7 para la instancia de tiempo T1 y así sucesivamente.

[0081] Para propósitos de brevedad, la divulgación puede usar las siguientes definiciones:

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

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

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

[0082] En la FIG. 2, cada una de las vistas incluye conjuntos de imágenes. Por ejemplo, la vista S0 incluye un conjunto de imágenes 0, 8, 16, 24, 32, 40, 48, 56 y 64, la vista S1 incluye el conjunto de imágenes 1, 9, 17, 25, 33, 41, 49, 57 y 65, y así sucesivamente. Para la codificación de vídeo 3D-HEVC, por ejemplo, cada imagen puede incluir dos imágenes componente: una imagen componente se denomina componente de vista de textura y la otra imagen componente se denomina componente de vista de profundidad. El componente de textura de vista y el componente de profundidad de vista dentro de un conjunto de imágenes de una vista se pueden considerar como correspondientes entre sí. Por ejemplo, el componente de textura de vista dentro de un conjunto de imágenes de una vista se considera correspondiente al componente de profundidad de vista dentro del conjunto de las imágenes de la vista, y viceversa (es decir, el componente de profundidad de vista corresponde a su componente de textura de vista en el conjunto, y viceversa). Como se usa en esta divulgación, un componente de vista de textura que corresponde a un componente de vista de profundidad se puede considerar como el componente de vista de textura, y el componente de vista de profundidad como parte de una misma vista de una única unidad de acceso.

[0083] El componente de vista de textura incluye el contenido de la imagen real que se visualiza. Por ejemplo, el componente de textura de la vista puede incluir los componentes de luma (Y) y croma (Cb y Cr). El componente de vista de profundidad puede indicar profundidades relativas de los píxeles en su correspondiente componente de vista de textura. A modo de ejemplo, el componente de profundidad de la vista es una imagen en escala de grises que incluye solamente valores de luma. En otras palabras, el componente de vista de profundidad puede que no transmita ningún contenido de imagen, sino que, en cambio, proporcione una cierta medida de las profundidades relativas de los píxeles en el componente de vista de textura.

[0084] Por ejemplo, un píxel blanco puro en el componente de vista de profundidad indica que su correspondiente píxel o píxeles en el componente de vista de textura correspondiente está más cerca de la perspectiva del espectador, y un píxel negro puro en el componente de vista de profundidad indica que su correspondiente píxel o píxeles en el componente de vista de textura correspondiente está más alejado de la perspectiva del espectador. Los diversos tonos de gris entre negro y blanco indican diferentes niveles de profundidad. Por ejemplo, un píxel muy gris en el componente de profundidad de la vista indica que su píxel correspondiente en el componente de textura de la vista está más alejado que un píxel ligeramente gris en el componente de profundidad de la vista. Dado que solamente es necesaria la escala de grises para identificar la profundidad de los píxeles, el componente de profundidad de la vista no necesita incluir componentes de croma, ya que los valores de color para el componente de profundidad de la vista pueden no tener ningún sentido.

[0085] El componente de vista de profundidad que usa solamente valores de luma (por ejemplo, valores de intensidad) para identificar la profundidad se proporciona con propósitos ilustrativos y no se debe considerar limitante. En otros ejemplos, se puede utilizar cualquier técnica para indicar las profundidades relativas de los píxeles en el componente de textura de la vista.

[0086] En la FIG. 3, se muestra una estructura de predicción MVC típica (que incluye tanto la predicción entre imágenes dentro de cada predicción de vista e entrevista) para la codificación de vídeo multivista. Las direcciones de predicción se indican mediante flechas, el objeto apuntado, usando el objeto desde el que se apunta como la referencia de predicción. En la MVC, la predicción entrevista se apoya en la compensación de movimiento por disparidad, que usa la sintaxis de la compensación de movimiento de H.264/AVC, pero permite que una imagen en una vista diferente sea usada como imagen de referencia.

[0087] En el ejemplo de la FIG. 3, se ilustran ocho vistas (que tienen ID de vista de "S0" hasta "S7"), y se ilustran doce localizaciones temporales (de "T0" hasta "T11") para cada vista. Es decir, cada fila en la FIG. 3 corresponde a una vista, mientras que cada columna indica una localización temporal.

[0088] Aunque la MVC tiene una denominada vista básica que se puede descodificar mediante los descodificadores de la H.264/AVC y se podrían admitir también los pares de vistas en estéreo por la MVC, la ventaja de MVC es que podría admitir un ejemplo que usa más de dos vistas como una entrada de vídeo 3D y que descodifica este vídeo 3D representado por las múltiples vistas. Un visualizador de un cliente que tiene un descodificador de MVC puede esperar contenido de vídeo en 3D con múltiples vistas.

[0089] Las imágenes en la FIG. 3 se indican en la intersección de cada fila y cada columna. La norma H.264/AVC puede usar el término trama para representar una porción del vídeo. Esta divulgación puede usar los términos imagen y trama indistintamente.

[0090] Las imágenes de la FIG. 3 se ilustran usando un bloque que incluye una letra, designando la letra si la imagen correspondiente está intracodificada (es decir, una imagen I), o intercodificada en una dirección (es decir, como una imagen P) o en múltiples direcciones (es decir, como una imagen B). En general, las predicciones se indican mediante flechas, donde las imágenes a las que se apunta usan la imagen desde la que se apunta como referencia de predicción. Por ejemplo, la imagen P de la vista S2 en la localización temporal T0 se predice a partir de la imagen I de la vista S0 en la localización temporal T0.

[0091] Al igual que con la codificación de vídeo de vista única, las imágenes de una secuencia de vídeo de codificación de vídeo de multivista se pueden codificar predictivamente con respecto a las imágenes en diferentes localizaciones temporales. Por ejemplo, la imagen b de la vista S0 en la localización temporal T1 tiene una flecha apuntando a la misma desde la imagen I de la vista S0 en la localización temporal T0, indicando que la imagen b se predice a partir de la imagen I. Adicionalmente, sin embargo, en el contexto de la codificación de vídeo multivista, las imágenes se pueden predecir entrevista. Es decir, un componente de la vista puede usar los componentes de la vista en otras vistas como referencia. En MVC, por ejemplo, la predicción entrevista se realiza como si el componente de vista en otra vista fuera una referencia de interpredicción. Las posibles referencias entrevista se señalizan en la extensión de la MVC del conjunto de parámetros de secuencia (SPS) y pueden ser modificadas por el proceso de construcción de la lista de imágenes de referencia, que habilita el ordenamiento flexible de las referencias de interpredicción o predicción entrevista. La predicción entrevista también es un rasgo característico de una extensión multivista propuesta de la HEVC, que incluye la 3D-HEVC (multivista más profundidad).

[0092] La FIG. 3 proporciona diversos ejemplos de predicción entrevista. Las imágenes de la vista S1, en el ejemplo de la FIG. 3, se ilustran como predichas a partir de imágenes en diferentes localizaciones temporales de la vista S1, así como predichas entrevista a partir de imágenes de las vistas S0 y S2 en las mismas localizaciones temporales. Por ejemplo, la imagen b de la vista S1 en la localización temporal T1 se predice a partir de cada una de las imágenes B de la vista S1 en las localizaciones temporales T0 y T2, así como las imágenes b de las vistas S0 y S2 en la localización temporal T1.

[0093] En algunos ejemplos, la FIG. 3 se puede ver como una ilustración de los componentes de vista de textura. Por ejemplo, las imágenes I, P, B y b, ilustradas en la FIG. 2, se pueden considerar componentes de vista de textura para cada una de las vistas. De acuerdo con las técnicas descritas en esta divulgación, para cada uno de los componentes de textura de la vista ilustrados en la FIG. 3 existe un componente de profundidad de la vista correspondiente. En algunos ejemplos, los componentes de profundidad de la vista se pueden predecir de una manera similar a la ilustrada en la FIG. 3 para los componentes de textura de la vista correspondientes.

[0094] La codificación de dos vistas también se puede admitir en MVC. Una de las ventajas de la MVC es que un codificador de MVC podría tomar más de dos vistas como una entrada de vídeo 3D y un descodificador de MVC puede descodificar tal representación de multivista. Por lo tanto, cualquier visualizador con un descodificador MVC puede esperar contenidos de vídeo 3D con más de dos vistas.

[0095] En MVC, se permite la predicción intervista entre imágenes en la misma unidad de acceso (es decir, con la misma instancia de tiempo). Cuando se codifica una imagen de una de las vistas no base, se puede añadir una imagen a una lista de imágenes de referencia si está en una vista diferente pero presenta una misma instancia de tiempo. Una imagen de referencia intervista se puede disponer en cualquier posición de una lista de imágenes de referencia, tal como cualquier imagen de referencia interpredicción. Como se muestra en la FIG. 3, un componente de la vista puede usar los componentes de la vista en otras vistas como referencia. En la MVC, la predicción intervista se realiza como si el componente de vista en otra vista fuera una referencia de interpredicción.

[0096] En el contexto de codificación de vídeo multivista, en general existen dos tipos de vectores de movimiento. Uno se denomina vector de movimiento normal. Los vectores de movimiento normales apuntan a imágenes de referencia temporal, y el correspondiente modo de interpredicción es la predicción con compensación de movimiento (MCP). El otro vector de movimiento es un vector de movimiento de disparidad (DMV). El DMV apunta a imágenes en una vista diferente (es decir, imágenes de referencia intervista) y la interpredicción correspondiente es la predicción con compensación de disparidad (DCP).

[0097] Otro tipo de formato de codificación de vídeo multivista introduce el uso de valores de profundidad (por ejemplo, como en 3D-HEVC). Para el formato de datos multivista-video-más-profundidad (MVD), que es popular para televisión 3D y videos de puntos de vista libres, las imágenes de textura y los mapas de profundidad se pueden codificar con imágenes de textura multivista de forma independiente. La FIG. 4 ilustra el formato de datos MVD con una imagen de textura y su mapa de profundidad por muestra asociado. El intervalo de profundidad puede estar restringido para estar en el intervalo de distancia mínima z_{near} y máxima z_{far} desde la cámara para los puntos 3D correspondientes.

[0098] En HEVC, las técnicas para la predicción de vector de movimiento pueden incluir un modo de fusión, un modo de omisión y un modo de predicción de vector de movimiento anticipado (AMVP). En general, de acuerdo con el modo de fusión y/o modo de omisión, un bloque de vídeo actual (por ejemplo, una PU) hereda la información de movimiento, por ejemplo, vector de movimiento, dirección de predicción e índice de imagen de referencia, de otro bloque contiguo previamente codificado, por ejemplo, un bloque espacialmente contiguo en la misma imagen, o un bloque en una imagen de referencia temporal o intervista. Cuando se implementa el modo de fusión/omisión, el codificador de vídeo 20 construye una lista de candidatos de fusión que son la información de movimiento de los bloques de referencia en una materia definida, selecciona uno de los candidatos de fusión y señala un índice de lista de candidatos que identifica al candidato de fusión seleccionado para descodificador de vídeo 30 en el flujo de bits.

[0099] El descodificador de vídeo 30, al implementar el modo de fusión/omisión, recibe este índice de lista de candidatos, reconstruye la lista de candidatos de fusión de acuerdo con la manera definida y selecciona uno de los candidatos de fusión en la lista de candidatos indicada por el índice. El descodificador de vídeo 30 puede entonces usar el candidato seleccionado de los candidatos de fusión como un vector de movimiento para la PU actual con la misma resolución que el vector de movimiento del candidato seleccionado de los candidatos de fusión y apuntando a la misma imagen de referencia que el vector de movimiento para el seleccionado de los candidatos de fusión. El modo de fusión y el modo de omisión promueven la eficacia del flujo de bits permitiendo que el codificador de vídeo 20 señale un índice en la lista de candidatos de fusión, en lugar de toda la información de movimiento para la interpredicción del bloque de vídeo actual.

[0100] Cuando se implementa AMVP, el codificador de vídeo 20 construye una lista de predictores de vector de movimiento candidatos (MVP) en una materia definida, selecciona uno de los MVP candidatos y señala un índice de lista de candidatos que identifica el MVP seleccionado con el descodificador de vídeo 30 en el flujo de bits. Similar al modo de fusión, al implementar AMVP, el descodificador de vídeo 30 reconstruye la lista de MVP candidatos en el asunto definido, y selecciona uno de los MVP en base al índice de lista de candidatos.

[0101] Sin embargo, contrariamente al modo de fusión/omisión, al implementar AMVP, el codificador de vídeo 20 también señala un índice de imagen de referencia y una dirección de predicción, especificando así la imagen de referencia a la que apunta el MVP especificado por los puntos del índice de lista de candidatos. Además, el codificador de vídeo 20 determina una diferencia de vector de movimiento (MVD) para el bloque actual, donde la MVD es una diferencia entre el MVP y el vector de movimiento real que de otro modo se usaría para el bloque actual. Para AMVP, además del índice de imagen de referencia, la dirección de la imagen de referencia y el índice de lista de candidatos, el codificador de vídeo 20 señala la MVD para el bloque actual en el flujo de bits. Debido a la señalización del índice de imagen de referencia y la diferencia de vector de predicción para un bloque dado, el AMVP puede no ser tan eficaz como el modo de fusión/omisión, pero puede proporcionar una fidelidad mejorada de los datos de vídeo codificados.

[0102] La FIG. 5 muestra un ejemplo de un bloque de vídeo actual 47, cinco bloques contiguos espaciales (41, 42, 43, 44 y 45) y un bloque de referencia temporal 46 de otra imagen pero en la misma vista que la imagen actual. El bloque de referencia temporal 46 puede ser, por ejemplo, un bloque coubicado en una imagen de una instancia temporal diferente, pero la misma vista que el bloque de vídeo actual 47. En algunos ejemplos, el bloque de vídeo actual 47 y los bloques de vídeo de referencia 41-46 se pueden definir en general en la norma HEVC actualmente en desarrollo. Los bloques de vídeo de referencia 41-46 se marcan A0, A1, B0, B1, B2 y T de acuerdo con la norma HEVC. El codificador de vídeo 20 y el descodificador de vídeo 30 pueden predecir la información de movimiento, incluyendo un TMV, del bloque de vídeo actual 47 en base a la información de movimiento de los bloques de vídeo de referencia 41-

46 de acuerdo con un modo de predicción de información de movimiento, por ejemplo, modo de fusión/omisión o modo AMVP. Como se describe con más detalle a continuación, los TMV de bloques de vídeo se pueden usar con DMV para predicción residual avanzada de acuerdo con las técnicas de esta divulgación.

[0103] Como se ilustra en la FIG. 5, los bloques de vídeo 42, 44, 43, 41 y 45 pueden estar a la izquierda, por encima, por encima a la derecha, por debajo a la izquierda y por encima a la izquierda, respectivamente, en relación con el bloque de vídeo actual 47. Sin embargo, el número y las localizaciones de los bloques contiguos 41-45 con respecto al bloque de vídeo actual 47 ilustrado en la FIG. 5 son solo ejemplos. La información de movimiento de un número diferente de bloques contiguos y/o de bloques en localizaciones diferentes, se puede considerar para su inclusión en una lista de candidatos de predicción de información de movimiento para el bloque de vídeo actual 47.

[0104] La relación espacial de cada uno de los bloques espacialmente contiguos 42, 44, 43, 41 y 45 con el bloque de vídeo actual 47 se puede describir de como sigue. Se usa una localización de luma (x_P , y_P) para especificar la muestra de luma superior izquierda del bloque actual en relación con la muestra superior izquierda de la imagen actual. Las variables $nPSW$ y $nPSH$ indican la anchura y la altura del bloque actual para luma. La muestra de luma superior izquierda del bloque 42 espacialmente contiguo es $x_P - 1$, $y_P + nPSH - 1$. La muestra de luma superior izquierda del bloque 44 espacialmente contiguo es $x_P + nPSW - 1$, $y_P - 1$. La muestra de luma superior izquierda del bloque espacialmente contiguo 43 es $x_P + nPSW$, $y_P - 1$. La muestra de luma superior izquierda del bloque espacialmente contiguo 41 es $x_P - 1$, $y_P + nPSH$. La muestra de luma superior izquierda del bloque espacialmente contiguo 45 es $x_P - 1$, $y_P - 1$. Aunque se describen con respecto a las localizaciones de luma, los bloques actuales y de referencia pueden incluir componentes de croma.

[0105] Cada uno de los bloques 41-45 espacialmente contiguos puede proporcionar un candidato de información de movimiento espacial para predecir la información de movimiento, por ejemplo, TMV, del bloque de vídeo actual 47. Un codificador de vídeo, por ejemplo, el codificador de vídeo 20 (FIG. 1) y/o descodificador de vídeo 30 (FIG. 1), puede considerar la información de movimiento de los bloques de referencia espacialmente contiguos en un orden predeterminado, por ejemplo, un orden de exploración. En el caso de 3D-HEVC, por ejemplo, el descodificador de vídeo puede considerar la información de movimiento de los bloques de referencia para su inclusión en una lista de candidatos de fusión para el modo de fusión en el siguiente orden: 42, 44, 43, 41 y 45. En el ejemplo ilustrado, los bloques espacialmente contiguos 41-45 están a la izquierda de y/o encima del bloque de vídeo actual 47. Esta disposición es típica, ya que la mayoría de los codificadores de vídeo codifican bloques de vídeo en orden de escaneado de trama desde la parte superior izquierda de una imagen. Por consiguiente, en dichos ejemplos, los bloques espacialmente contiguos 41 - 45 se codificarán típicamente antes del bloque de vídeo actual 47. Sin embargo, en otros ejemplos, por ejemplo, cuando un codificador de vídeo codifica bloques de vídeo en un orden diferente, los bloques 41-45 espacialmente contiguos pueden estar localizados a la derecha de y/o debajo del bloque de vídeo actual 47.

[0106] El bloque de referencia temporal 46 está situado dentro de una imagen de referencia temporal codificada previamente, aunque no necesariamente inmediatamente antes en orden de codificación, a la imagen actual del bloque de vídeo actual 47. Además, la imagen de referencia del bloque 46 no es necesariamente anterior a la imagen del bloque de vídeo actual 47 en orden de vista. El bloque de vídeo de referencia 46 puede estar cubierto, en general, en la imagen de referencia con respecto a la localización del bloque de vídeo actual 47 en la imagen actual. En algunos ejemplos, el bloque de vídeo de referencia 46 está situado a la derecha de y por debajo de la posición del bloque de vídeo actual 47 en la imagen actual o cubre la posición central del bloque de vídeo actual 47 en la imagen actual.

[0107] La FIG. 6 es un diagrama conceptual que ilustra un ejemplo de obtención de un candidato de vector de movimiento predicho de entrevista (IPMVC) y un candidato de vector de movimiento de disparidad de entrevista (IDMVC) para la predicción de la información de movimiento de un bloque de vídeo actual 50, por ejemplo de acuerdo con un modo de fusión/omisión o un modo AMVP. Cuando la predicción de entrevista está habilitada, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden añadir un nuevo candidato de vector de movimiento, IPMVC o IDMVC a la lista de candidatos de información de movimiento para el bloque de vídeo actual 50. El IPMVC puede predecir un TMV para el bloque de vídeo actual 50, que el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden usar para ARP del bloque de vídeo actual 50 u otro bloque de vídeo de acuerdo con las técnicas de esta divulgación, como se describe con mayor detalle a continuación. El IDMVC puede predecir un DMV para el bloque de vídeo actual 50, que el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden usar para ARP del bloque de vídeo actual 50.

[0108] En el ejemplo de la FIG. 6, el bloque actual 50 está en la vista actual V_m . El codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden localizar un bloque 52 correspondiente o de referencia en la vista de referencia V_0 usando un vector de disparidad (DV) 51. El codificador de vídeo puede determinar DV 51 en base a parámetros de cámara, o de acuerdo con cualquiera de las técnicas descritas en el presente documento. Por ejemplo, el codificador de vídeo puede determinar DV 51 para el bloque de vídeo actual 50 incluyendo en base a un DMV o DV de un bloque contiguo, por ejemplo, usando la obtención de vector de disparidad basado en bloques contiguos (NBDV).

[0109] Si el bloque de referencia 52 no está intracodificado y no tiene predicción de entrevista, y su imagen de referencia, por ejemplo, la imagen de referencia 58 o la imagen de referencia 60, tiene un valor de recuento de orden

de imagen (POC) igual al de una entrada en la misma lista de imágenes de referencia del bloque de vídeo actual 50, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden obtener su información de movimiento (dirección de predicción, imágenes de referencia y vectores de movimiento), después de convertir el índice de referencia en base a POC, para ser el IPMVC para el bloque de vídeo actual 50.

[0110] En el ejemplo de la FIG. 6, el bloque de vídeo de referencia 52 está asociado con TMV 54 que apunta a una primera imagen de referencia 58 en la vista de referencia V0 que se especifica en una primera lista de imágenes de referencia (RefPicList0) y TMV 56 que apunta a una segunda imagen 60 en la vista de referencia V0 que se especifica en una segunda lista de imágenes de referencia (RefPicList1). La herencia de los TMV 54 y 56 por el bloque de vídeo actual 50 se ilustra mediante las flechas discontinuas en la FIG.6. En base a la información de movimiento del bloque de vídeo de referencia 52, el codificador de vídeo obtiene el IPMVC para que el bloque de vídeo actual 50 sea al menos uno de un TMV 62 que apunte a una primera imagen de referencia 66 en la vista actual Vm que se especifica en una primera lista de imágenes de referencia (RefPicList0), por ejemplo, con el mismo POC en la primera lista de imágenes de referencia como imagen de referencia 58 y TMV 64 apuntando a una segunda imagen 68 en la vista actual Vm que se especifica en una segunda lista de imágenes de referencia (RefPicList1) por ejemplo, con el mismo POC que la imagen de referencia 60.

[0111] El codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden usar TMV 62 y/o TMV 64 para ARP del bloque de vídeo actual 50. El codificador de vídeo 20 y/o decodificador de vídeo 30 también pueden convertir DV 51 a un IDMVC para el bloque de vídeo actual 50 y añadir el IDMVC a la lista de candidatos de información de movimiento para el bloque de vídeo actual 50 en una posición diferente de la IPMVC. Cada uno de los IPMVC o IDMVC se puede denominar un "candidato entrevista" en este contexto.

[0112] En el modo de fusión/omisión, el codificador de vídeo inserta el IPMVC, si está disponible, antes de todos los candidatos de fusión espacial y temporal en la lista de candidatos de fusión. En el modo de fusión/omisión, el codificador de vídeo inserta el IDMVC antes del candidato de fusión espacial derivado de A0 (bloque 41 de la FIG. 5). La conversión de DV 51 a un IDMVC se puede considerar conversión de DV 51 a un DMV para el bloque de vídeo actual 50. El codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden usar DMV para ARP del bloque de vídeo actual 50.

[0113] En algunas situaciones, un codificador de vídeo puede obtener un DV para un bloque de vídeo actual. Por ejemplo, como se describe anteriormente en referencia a la FIG. 6, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden obtener DV 51 para el bloque de vídeo actual 50. En algunos ejemplos, un codificador de vídeo puede usar la obtención NBDV para obtener el DV para el bloque de vídeo actual. La obtención NBDV se usa como procedimiento de obtención de vectores de disparidad en 3D-HEVC.

[0114] Las propuestas para 3D-HEVC usan un orden de codificación de textura primero para todas las vistas. En otras palabras, para cada una de la pluralidad de vistas en el flujo de bits, el componente de textura de está codificado, por ejemplo, codificado o decodificado, antes de cualquier componente de profundidad para la vista. En algunos casos, por ejemplo, para la predicción de entrevista, se necesita un DV para codificar un bloque de vídeo en un componente de textura de una vista en una unidad de acceso particular. Sin embargo, en la codificación de texto primero, el componente de profundidad correspondiente del bloque de vídeo actual no está disponible para determinar el DV para el bloque de vídeo actual. La obtención NBDV puede ser empleado por un codificador de vídeo, y se propone para 3D-HEVC, para obtener un DV para un bloque de vídeo actual en dichas situaciones. En el diseño actual de 3D-HEVC, el DV obtenido de la obtención NBDV se podría refinar adicionalmente recuperando los datos de profundidad del mapa de profundidad de la vista de referencia apuntados por el DV desde el proceso NBDV.

[0115] Un DV se usa para un estimador del desplazamiento entre dos vistas. Debido a que los bloques contiguos comparten casi la misma información de movimiento/disparidad en la codificación de vídeo, el bloque de vídeo actual puede usar la información de vector de movimiento en bloques contiguos como un buen predictor de su información de movimiento/disparidad. Siguiendo esta idea, la obtención NBDV usa la información de disparidad contigua para estimar los DV en diferentes vistas.

[0116] De acuerdo con la obtención NBDV, un codificador de vídeo identifica varios bloques contiguos espaciales y temporales. Se utilizan dos conjuntos de bloques contiguos. Un conjunto es de bloques contiguos espacialmente y el otro conjunto es de bloques contiguos temporalmente. El codificador de vídeo comprueba cada uno de los bloques contiguos espaciales y temporales en un orden predefinido determinado por la prioridad de la correlación entre el bloque actual y el bloque candidato (contiguo). Cuando el codificador de vídeo identifica un DMV, es decir, un vector de movimiento que apunta desde el bloque candidato contiguo a una imagen de referencia entrevista (en la misma unidad de acceso, pero en una vista diferente), en la información de movimiento de los candidatos, el codificador de vídeo convierte el DMV a un DV, y devuelve el índice de orden de vista asociado. Por ejemplo, el codificador de vídeo puede establecer un componente horizontal del DV para el bloque actual igual a un componente horizontal del DMV y puede establecer el componente vertical del DV en 0.

[0117] 3D-HEVC adoptó inicialmente las técnicas de obtención NBDV propuestas en Zhang et al. "3D-CE5.h: Disparity vector generation results", Equipo de Colaboración Conjunta en el Desarrollo de la Extensión de Codificación de Vídeo

de la ITU-T SG 16 WP 3 e ISO/IEC JTC1/SC 29/WG 11, 1ª Reunión: Estocolmo, Suecia, 16-20 de julio de 2012, Doc. JCT3V-A0097 (número MPEG m26052), a continuación en el presente documento "JCT3V-A0097". JCT3V-A0097 se puede descargar desde el siguiente enlace: http://phenix.int-evry.fr/jct2/doc_end_user/current_document.php?id=89.

[0118] En algunas propuestas de 3D-HEVC, cuando el codificador de vídeo realiza el proceso de obtención NBDV, el codificador de vídeo comprueba los vectores de movimiento de disparidad en los bloques contiguos temporales, los vectores de movimiento de disparidad en los bloques contiguos espaciales y luego los vectores de disparidad implícitos (IDVs) en orden. Un IDV puede ser un vector de disparidad de una PU espacialmente o temporalmente contigua que se codifica usando la predicción intervista. Los IDV también se pueden denominar vectores de disparidad derivados. Un IDV se puede generar cuando una PU emplea la predicción de entrevista, es decir, el candidato para AMVP o modos de fusión se obtiene a partir de un bloque de referencia en la otra vista con la ayuda de un vector de disparidad. Dicho vector de disparidad se llama IDV. Un IDV se puede almacenar en la PU con el propósito de obtención DV. Por ejemplo, aunque el bloque se codifica con predicción de movimiento, un DV derivado para el bloque no se descarta con el propósito de codificar un bloque de vídeo siguiente. Por lo tanto, cuando el codificador de vídeo identifica un DMV o un IDV, el codificador de vídeo puede devolver el DMV o IDV identificado.

[0119] Se incluyeron vectores de disparidad implícitos (IDV) con un proceso de obtención NBDV simplificado descrito en Sung et al., "3D-CE5.h: Simplification of disparity vector derivation for HEVC-based 3D video coding", Equipo de Colaboración Conjunta en el Desarrollo de la Extensión de Codificación de Vídeo de la ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 1ª Reunión: Estocolmo, Suecia, 16-20 de julio de 2012, Doc. JCT3V-A0126 (número MPEG m26079), a continuación en el presente documento "JCT3V-A0126". JCT3V-A0126 se puede descargar desde el siguiente enlace: http://phenix.int-evry.fr/jct2/doc_end_user/current_document.php?id=142.

[0120] El desarrollo adicional de un proceso de obtención de NBDV para 3D-HEVC se describió en Kang et al., "3D-CE5.h: Improvements for disparity vector derivation", Equipo de Colaboración Conjunta sobre Elaboración de Extensión de Codificación de Vídeo de la ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 2ª Reunión: Shanghái, China, 13-19 de octubre de 2012, Doc. JCT3V-B0047 (número MPEG m26736), a continuación en el presente documento "JCT3V-B0047". JCT3V-B0047 se puede descargar desde el siguiente enlace: http://phenix.it-sudparis.eu/jct2/doc_end_user/current_document.php?id=236.

[0121] En JCT3V-B0047, el proceso de obtención NBDV para 3D-HEVC se simplificó aún más retirando los IDV almacenados en la memoria intermedia de imágenes descodificadas. La ganancia de codificación también mejoró con la selección de imágenes del punto de acceso aleatorio (RAP). El codificador de vídeo puede convertir el IDV o vector de movimiento de disparidad devuelto en un vector de disparidad y puede usar el vector de disparidad para la predicción de entrevista y la predicción residual de entrevista. El acceso aleatorio se refiere a la descodificación de un flujo de bits a partir de una imagen codificada que no es la primera imagen codificada en el flujo de bits. La inserción de imágenes de acceso aleatorio o puntos de acceso aleatorio en un flujo de bits a intervalos regulares puede permitir el acceso aleatorio. Entre los tipos de ejemplo de imágenes de acceso aleatorio se incluyen imágenes de actualización instantánea de descodificador instantánea (IDR), imágenes de acceso aleatorio limpio (CRA) e imágenes de acceso de enlace roto (BLA). Por lo tanto, las imágenes IDR, CRA y BLA se denominan colectivamente imágenes RAP. En algunos ejemplos, las imágenes RAP pueden tener tipos de unidades NAL iguales a BLA_W_LP, BLA_W_RADL, BLA_N_LP, IDR_W_RADL, IDR_N_LP, RSV_IRAP_VCL22, RSV_IRAP_VCL23, o CRA_NUT.

[0122] Las técnicas para la obtención DV basado en CU para 3D-HEVC se propusieron en Kang et al., "CE2.h: CU-based disparity vector derivation in 3D-HEVC", Equipo de Colaboración Conjunta en el Desarrollo de la Extensión de Codificación de Vídeo de la ITU-T SG 16 WP3 e ISO/IEC JTC 1/SC 29/WG 11, 4ª Reunión: Incheon, Corea, 20-26 de abril de 2013, Doc. JCT3V-D0181 (número MPEG m29012), a continuación en el presente documento "JCT3V-D0181". JCT3V-D0181 se puede descargar desde el siguiente enlace: http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=866.

[0123] Cuando el codificador de vídeo identifica un DMV o un IDV, el codificador de vídeo puede terminar el proceso de comprobación. Así, una vez que el codificador de vídeo encuentra un DV para el bloque actual, el codificador de vídeo puede terminar el proceso de obtención NBDV. Cuando el codificador de vídeo no puede determinar un DV para el bloque actual realizando el proceso de obtención NBDV (es decir, cuando no se encuentra DMV o IDV durante el proceso de obtención NBDV), el NBDV está marcado como no disponible. En otras palabras, se puede considerar que el proceso de obtención NBDV devuelve un vector de disparidad no disponible.

[0124] Si el codificador de vídeo no puede obtener un DV para el bloque actual (es decir, si no se encuentra ningún vector de disparidad) realizando el proceso de obtención NBDV, el codificador de vídeo puede usar un DV cero como DV para la PU actual. El DV cero es un DV que tiene componentes horizontales y verticales iguales a 0. Por tanto, incluso cuando el proceso de obtención NBDV devuelve un resultado no disponible, otros procesos de codificación del codificador de vídeo que requieren un DV pueden usar un vector de disparidad cero para el bloque actual. En algunos ejemplos, si el codificador de vídeo es incapaz de obtener un DV para el bloque actual realizando el proceso de obtención NBDV, el codificador de vídeo puede deshabilitar la predicción residual de entrevista para el bloque actual. Sin embargo, independientemente de si el codificador de vídeo es capaz de obtener un DV para el bloque actual realizando el proceso de obtención NBDV, el codificador de vídeo puede usar la predicción de entrevista para el bloque

actual. Es decir, si no se encuentra un DV después de comprobar todos los bloques contiguos predefinidos, se puede usar un vector de disparidad cero para la predicción de entrevista mientras que la predicción residual de entrevista puede estar deshabilitada para la CU correspondiente.

[0125] La FIG. 7 es un diagrama conceptual que ilustra ejemplos de bloques contiguos espaciales, a partir de los cuales se puede obtener un DV para un bloque de vídeo actual usando la obtención NBDV, con relación al bloque de vídeo actual 90. Los cinco bloques contiguos espaciales en la FIG. 7 son el bloque inferior izquierdo 96, el bloque izquierdo 95, el bloque superior derecho 92, el bloque superior 93 y el bloque superior izquierdo 94 en relación con el bloque de vídeo actual. Los bloques contiguos espaciales pueden ser los bloques inferior izquierdo, izquierdo, superior-derecho, superior y superior izquierdo de un bloque de vídeo actual que cubre CU. Se debe observar que estos bloques contiguos espaciales para NBDV pueden ser los mismos que los bloques contiguos espaciales usados por el codificador de vídeo para la predicción de información de movimiento para el bloque de vídeo actual, tal como de acuerdo con los modos MERGE/AMVP en HEVC. En dichos casos, no se requiere ningún acceso adicional a la memoria por parte del codificador de vídeo para NBDV, ya que la información de movimiento de los bloques contiguos espaciales ya se considera para la predicción de información de movimiento para el bloque de vídeo actual.

[0126] Para comprobar bloques contiguos temporales, un codificador de vídeo construye una lista de imágenes candidatas. En algunos ejemplos, el codificador de vídeo puede tratar hasta dos imágenes de referencia desde la vista actual, es decir, la misma vista que el bloque de vídeo actual, como imágenes candidatas. El codificador de vídeo puede insertar primero la imagen de referencia coubicada en la lista de imágenes candidatas, seguida por el resto de imágenes candidatas en el orden ascendente del índice de imágenes de referencia. Cuando estén disponibles imágenes de referencia con el mismo índice de referencia en ambas listas de imágenes de referencia, el codificador de vídeo puede insertar la de la misma lista de imágenes de referencia que la imagen coubicada antes de la otra imagen de referencia de la otra lista de imágenes de referencia. En algunos ejemplos, el codificador de vídeo puede identificar tres regiones candidatas para obtener los bloques contiguos temporales de cada una de las imágenes candidatas en la lista de imágenes candidatas. Las tres regiones candidatas se pueden definir como sigue:

- CPU: Una región coubicada de la PU actual o la CU actual.
- CLCU: Una unidad de codificación más grande (LCU) que cubre la región coubicada del bloque actual.
- BR: Un bloque inferior derecho 4x4 de la CPU.

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

[0127] Como se analizó anteriormente, además de los DMV derivados de bloques contiguos espaciales y temporales, un codificador de vídeo puede comprobar IDV. En el proceso de obtención NBDV propuesto para 3D-HTM 7.0 y versiones posteriores, el codificador de vídeo comprueba DMV en los bloques contiguos temporales, luego los DMV en los bloques contiguos espaciales, y luego los IDV, en orden. Una vez que se encuentra el DMV o IDV, se termina el proceso. Además, el número de bloques contiguos espaciales verificados en el proceso de obtención NBDV se reduce aún más hasta dos.

[0128] Cuando el codificador de vídeo comprueba una PU contigua (es decir, una PU espacialmente o temporalmente contigua), el codificador de vídeo puede comprobar primero si la PU contigua tiene un vector de movimiento de disparidad. Si ninguna de las PU contiguas tiene un vector de movimiento de disparidad, el codificador de vídeo puede determinar si cualquiera de las PU espacialmente contiguas tiene un IDV. Si una de las PU espacialmente contiguas tiene un IDV y el IDV está codificado como modo de fusión/omisión, el codificador de vídeo puede terminar el proceso de comprobación y puede usar el IDV como vector de disparidad final para la PU actual.

[0129] Como se indicó anteriormente, un codificador de vídeo puede aplicar un proceso de obtención NBDV para obtener un DV para un bloque actual (por ejemplo, una CU, PU, etc.). El vector de disparidad para el bloque actual puede indicar una localización en una imagen de referencia (es decir, un componente de referencia) en una vista de referencia. En algunos diseños 3D-HEVC, el codificador de vídeo puede acceder a la información de profundidad para la vista de referencia. En algunos diseños 3D HEVC, cuando el codificador de vídeo usa el proceso de obtención NBDV para obtener el DV para el bloque actual, el codificador de vídeo puede aplicar un proceso de refinamiento para refinar aún más el vector de disparidad para el bloque actual. El codificador de vídeo puede refinar el DV para el bloque actual en base al mapa de profundidad de la imagen de referencia. El codificador de vídeo puede usar un proceso de refinamiento similar para refinar un DMV para la predicción de síntesis de vista hacia atrás. De esta manera, la profundidad se puede usar para refinar el DV o DMV que se va a usar para la predicción de síntesis de vista hacia atrás. Este proceso de refinamiento se puede denominar en este documento refinamiento de NBDV ("NBDV-R"), el proceso de refinamiento de NBDV, o NBDV orientado a la profundidad (Do-NBDV).

[0130] Cuando el proceso de obtención NBDV devuelve un vector de disparidad disponible (por ejemplo, cuando el proceso de obtención NBDV devuelve una variable que indica que el proceso de obtención NBDV pudo obtener un

vector de disparidad para el bloque actual en base a un vector de movimiento de disparidad o un IDV de un bloque contiguo), el codificador de vídeo puede refinar aún más el vector de disparidad recuperando los datos de profundidad del mapa de profundidad de la vista de referencia. En algunos ejemplos, el proceso de refinamiento incluye los dos pasos siguientes:

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

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

[0131] El DV refinado se puede usar para la predicción de entrevista para el bloque de vídeo actual, mientras que el DV no refinado se puede usar para la predicción residual de entrevista para el bloque de vídeo actual. Además, el DV refinado se almacena como el vector de movimiento de una PU si está codificado con el modo de predicción de síntesis de vista atrás (BVSP), que se describe con mayor detalle a continuación. En el proceso de NBDV propuesto para 3D-HTM 7.0 y versiones posteriores, se accede al componente de vista de profundidad de la vista de base independientemente del valor del índice de orden de vista derivado del proceso NBDV.

[0132] Se han propuesto técnicas de predicción de movimiento entrevista a nivel de sub-PU para generar un nuevo candidato de fusión en An et al., "3D-CE3: Sub-PU level inter-view motion prediction", Equipo de colaboración conjunta en el desarrollo de la extensión de codificación de vídeo 3D de la ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 6ª Reunión: Ginebra, Suiza, 25 de octubre -1 de noviembre 2013, Doc. JCT3V-F0110, a continuación en el presente documento "JCT3V-F0110". JCT3V-F0110 se puede descargar desde el siguiente enlace: http://phenix.it-sudparis.eu/jct2/doc_end_user/current_document.php?id=1447. El nuevo candidato se añade a la lista de candidatos de fusión.

[0133] La FIG. 8 es un diagrama conceptual que ilustra la predicción de movimiento entrevista de sub-unidad de predicción (PU) Como se muestra en la FIG. 8, la PU actual 98 en la vista actual V1 se puede dividir en múltiples sub-PU (por ejemplo, cuatro sub-PU). Se puede usar un vector de disparidad para cada sub-PU para localizar los bloques de referencia correspondientes en una vista de referencia V0. El codificador de vídeo 20 y/o el decodificador de vídeo 30 se pueden configurar para copiar (es decir, reutilizar) los vectores de movimiento asociados con cada uno de los bloques de referencia para su uso con las sub-PU correspondientes de la PU actual 8.

[0134] En un ejemplo, el nuevo candidato, denominado candidato de fusión sub-PU, se deriva usando las siguientes técnicas. En primer lugar, indicar el tamaño de la PU actual por nPSW x nPSH, el tamaño de la sub-PU señalizada por NxN y el tamaño de la sub-PU final por subWxsubH. La PU actual se puede dividir en una o múltiples sub-PU dependiendo del tamaño de la PU y el tamaño de la sub-PU señalizada como sigue:

$$\text{subW} = \max (N, \text{nPSW}) \neq N ? N : \text{nPSW};$$

$$\text{subH} = \max (N, \text{nPSH}) \neq N ? N : \text{nPSH};$$

[0135] El codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden establecer el vector de movimiento predeterminado tmvLX en (0, 0) y el índice de referencia reflX en -1 para cada lista de imágenes de referencia (donde X representa la lista de imágenes de referencia 0 o la lista de imágenes de referencia 1). Para cada uno de los sub-PU en orden de exploración ráster, se aplica lo siguiente.

- Añadir el DV obtenido de un proceso de obtención DoNBDV o proceso NBDV a la posición media de la sub-PU actual para obtener una localización de muestra de referencia (xRefSub, yRefSub) mediante:

$$\text{xRefSub} = \text{Clip3}(0, \text{PicWidthInSamplesL} - 1, \text{xPSub} + \text{nPSWsub}/2 + ((\text{mvDisp}[0] + 2) \gg 2))$$

$$\text{yRefSub} = \text{Clip3}(0, \text{PicHeightInSamplesL} - 1, \text{yPSub} + \text{nPSHsub}/2 + ((\text{mvDisp}[1] + 2) \gg 2))$$

Un bloque en la vista de referencia que abarca (xRefSub, yRefSub) se usa como el bloque de referencia para la sub-PU actual.

- Para el bloque de referencia identificado:

1) si el bloque de referencia identificado se codifica usando vectores de movimiento temporal, se aplica lo siguiente.

- Los parámetros de movimiento asociados se pueden usar como parámetros de movimiento candidatos para la sub-PU actual.
- $tmvLX$ y $refLX$ se actualizan con la información de movimiento de la sub-PU actual.
- Si la sub-PU actual no es la primera en el orden de barrido de trama, la información de movimiento ($tmvLX$ y $refLX$) es heredada por todas las sub-PU anteriores.

2) de otro modo (el bloque de referencia está intracodificado), la información de movimiento de la sub-PU actual se establece en $tmvLX$ y $refLX$.

[0136] Se pueden usar diferentes tamaños de bloques sub-PU en las técnicas descritas anteriormente para la predicción de movimiento inter vista a nivel de sub-PU, incluyendo 4x4, 8x8 y 16x16. El tamaño del bloque sub-PU se puede señalar en un conjunto de parámetros, tal como un conjunto de parámetros de vista (VPS).

[0137] La predicción residual avanzada (ARP) es una herramienta de codificación que busca explotar la correlación residual entre vistas para proporcionar una eficacia de codificación adicional. En la ARP, se produce un predictor residual al alinear la información de movimiento en la vista actual para la compensación de movimiento en la vista de referencia. Además, se introducen factores de ponderación para compensar las diferencias de calidad entre las vistas. Cuando la ARP está habilitada para un bloque, se señala la diferencia entre el residual actual y el predictor residual. Es decir, el residual de un bloque actual se resta del residual del predictor residual y se señala la diferencia resultante. En algunas propuestas para 3D-HEVC, solo se aplica ARP a CU intercodificadas con un modo de partición igual a $Part_2Nx2N$.

[0138] La FIG. 9 es un diagrama conceptual que ilustra un ejemplo de estructura de predicción para propuestas de ejemplo para ARP temporal de un bloque de vídeo predicho temporalmente. ARP aplicado a las CU con el modo de partición igual a $Part_2Nx2N$ fue adoptado en la 4.^a reunión de JCT3V, como se propone en Zhang et al., "CE4: Advanced residual prediction for multiview coding", Equipo de Colaboración Conjunta en el Desarrollo de la Extensión de Codificación de Vídeo de la ITU-T SG 16 WP3 e ISO/IEC JTC 1/SC 29/WG 11, 4^a Reunión: Incheon, Corea, 20-26 de abril de 2013, Doc. JCT3V-D0177 (número MPEG m29008), a continuación en el presente documento "JCT3V-D0177". JCT3V-D0177 se puede descargar desde el siguiente enlace: http://phenix.it-sudparis.eu/ict3v/doc_end_user/current_document.php?id=862_

[0139] Como se muestra en la FIG. 9, un codificador de vídeo invoca o identifica los siguientes bloques en la predicción del residual para un bloque de vídeo actual 100 en una imagen actual 102 de la vista actual V_m , por ejemplo, dependiente.

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

2) Un bloque de vídeo de referencia entrevista 106 en una imagen de referencia entrevista 108 de una vista de referencia/base (V_0 en la FIG. 9): **Base**. El codificador de vídeo obtiene el bloque de vídeo de referencia entrevista 106 en base a DV 104 del bloque de vídeo actual 100 (**Curr**). El codificador de vídeo puede determinar DV 104 usando la obtención NBDV, como se describe anteriormente.

3) Un bloque de vídeo de referencia temporal 112 en una imagen de referencia temporal 114 **en la misma vista** (V_m) que el bloque de vídeo actual 100 (**Curr**): **CurrTRef**. El codificador de vídeo obtiene el bloque de vídeo de referencia temporal 112 en base al TMV 110 del bloque de vídeo actual 100. El codificador de vídeo puede determinar TMV 110 usando cualquiera de las técnicas descritas en el presente documento.

4) Un bloque de vídeo de referencia temporal 116 en una imagen de referencia temporal 118 en la vista de referencia, es decir, la misma vista que el bloque de vídeo de referencia entrevista 106 (**Base**): **BaseTRef**. Un codificador de vídeo obtiene el bloque de vídeo de referencia temporal 116 en la vista de referencia usando TMV 110 del bloque de vídeo actual 100 (**Curr**). Un vector 121 de TMV+DV puede identificar el bloque de vídeo de referencia temporal 116 (**BaseTRef**) con respecto al bloque de vídeo actual 100 (**Curr**).

[0140] Cuando el codificador de vídeo 20 predice temporalmente el bloque de vídeo actual 100 en base al bloque de vídeo de referencia temporal 112, que el codificador de vídeo 20 identifica usando TMV 110, el codificador de vídeo 20 determina las diferencias de píxel a píxel entre el bloque de vídeo actual 100 y el bloque de vídeo de referencia temporal 112 como un bloque residual. Estando ausente ARP, el codificador de vídeo 20 transformaría, cuantificaría y codificaría por entropía el bloque residual. El decodificador de vídeo 30 decodificaría por entropía un flujo de bits de vídeo codificado, realizaría transformación y cuantificación inversa para obtener el bloque residual y aplicaría el bloque residual a una reconstrucción del bloque de vídeo de referencia 112 para reconstruir el bloque de vídeo actual 100.

[0141] Usando ARP, los codificadores de vídeo determinan un bloque predictor residual que predice los valores del bloque residual, es decir, predice la diferencia entre el bloque de vídeo actual 100 (**Curr**) y el bloque de vídeo de referencia temporal 112 (**CurrTRef**). El codificador de vídeo 20 puede entonces solamente necesitar codificar una diferencia entre el bloque residual y el bloque predictor residual, reduciendo la cantidad de información incluida en el flujo de bits de vídeo codificado para codificar el bloque de vídeo actual 100. En el ejemplo de ARP temporal de la FIG. 9, el predictor para el residual del bloque de vídeo actual 100 se determina en base a bloques en la vista de referencia/base (V_0) que corresponden al bloque de vídeo actual 100 (**Curr**) y al bloque de vídeo de referencia temporal 112 (**CurrTRef**), y se identifican mediante DV 104. La diferencia entre estos bloques correspondientes en la vista de referencia puede ser un buen predictor del residual, es decir, la diferencia entre el bloque de vídeo actual 100 (**Curr**) y el bloque de vídeo de referencia temporal 112 (**CurrTRef**). En particular, los codificadores de vídeo identifican un bloque de vídeo de referencia entrevista 106 (**Base**) y un bloque de vídeo de referencia temporal 116 (**BaseTRef**) en la vista de referencia y determinan el bloque predictor residual en base a la diferencia entre el bloque de vídeo de referencia entrevista 106 y el bloque de vídeo de referencia temporal 116 (**BaseTRef-Base**), en el que la operación de sustracción se aplica a cada píxel de las matrices de píxeles indicados. En algunos ejemplos, un codificador de vídeo puede aplicar un factor de ponderación, w , al predictor residual. En tales ejemplos, el predictor final del bloque actual, es decir, el bloque de referencia sumado con el bloque predictor residual, se puede designar como: **CurrTRef+ w *(BaseTRef-Base)**.

[0142] La FIG. 10 es un diagrama conceptual que ilustra un ejemplo de estructura de predicción bidireccional para ARP temporal de un bloque de vídeo actual 120 en una vista actual (V_m). Las descripciones anteriores y la FIG. 9 ilustran predicción unidireccional. Al extender ARP al caso de predicción bidireccional, el codificador de vídeo puede aplicar las técnicas anteriores a una o ambas listas de imágenes de referencia con el fin de identificar un bloque predictor residual para el bloque de vídeo actual 120. En particular, el codificador de vídeo puede comprobar una o ambas listas de referencia para el bloque de vídeo actual 120 para determinar si una de ellas contiene un TMV utilizable para ARP temporal. En el ejemplo ilustrado por la FIG. 10, el bloque de vídeo actual 120 está asociado con un TMV 130 que apunta a una primera imagen de referencia temporal 134 en una primera lista de imágenes de referencia (RefPicList0) y un TMV 132 que apunta a una segunda imagen de referencia temporal 136 en una segunda lista de imágenes de referencia (RefPicList1).

[0143] En algunos ejemplos, un codificador de vídeo comprobará las listas de imágenes de referencia de acuerdo con un orden de comprobación para determinar si una de ellas incluye un TMV utilizable para ARP temporal y no necesita comprobar una segunda lista de acuerdo con la orden de comprobación si una primera lista incluye un TMV de este tipo. En algunos ejemplos, un codificador de vídeo comprobará ambas listas de imágenes de referencia y, si ambas listas incluyen un TMV, determinará qué TMV usar, por ejemplo, en base a una comparación de los predictores residuales resultantes producidos usando los TMVs con respecto al residual del bloque de vídeo actual. En particular, de acuerdo con las propuestas actuales para ARP, por ejemplo JCT3VC-D0177, cuando el bloque actual usa una imagen de referencia entrevista (en una vista diferente) para una lista de imágenes de referencia, el proceso de predicción residual está deshabilitado.

[0144] Como se ilustra en la FIG. 10, el codificador de vídeo puede usar un DV 124 identificado para el bloque de vídeo actual 120, por ejemplo, de acuerdo con un proceso de obtención NBDV, para identificar un bloque de vídeo de referencia entrevista 126 (**Base**) en una imagen de referencia entrevista 128 que está en una vista de referencia diferente (V_0), pero está en la misma unidad de acceso que la imagen actual 122. El codificador de vídeo puede usar también los TMV 130 y 132 para que el bloque de vídeo actual 120 identifique los bloques de referencia temporales (**BaseTRef**) para el bloque de vídeo de referencia entrevista 126 (**Base**) en varias imágenes de referencia temporales de la vista de referencia en ambas listas de imágenes de referencia, por ejemplo, RefPicList0 y RefPicList1. En el ejemplo de la FIG. 10, el codificador de vídeo identifica el bloque de vídeo de referencia temporal (**BaseTRef**) 140 en la imagen de referencia temporal 142 en una primera lista de imágenes de referencia, por ejemplo, RefPicList0 y el bloque de vídeo de referencia temporal (**BaseTRef**) 144 en la imagen de referencia temporal 146 en una segunda lista de imágenes de referencia, por ejemplo, RefPicList1, en base a los TMV 130 y 132 del bloque de vídeo actual 120.

[0145] El uso de TMV 130 y 132 del bloque de vídeo actual 120 en la vista de referencia se ilustra mediante flechas discontinuas en la FIG. 10. En la FIG. 10, los bloques de vídeo de referencia temporales 140 y 144 en la vista de referencia se denominan bloques de referencia compensados por movimiento debido a su identificación en base a los TMV 130 y 132. Un codificador de vídeo puede determinar un bloque predictor residual para el bloque de vídeo actual 120 en base a una diferencia entre el bloque de vídeo de referencia temporal 140 y el bloque de vídeo de referencia entrevista 126, o en base a una diferencia entre el bloque de vídeo de referencia temporal 144 y el bloque de vídeo de referencia entrevista 126.

[0146] Para reiterar, el proceso ARP temporal propuesto en el lado del descodificador se puede describir (con referencia a la FIG. 10) como sigue:

1. El descodificador de vídeo 30 obtiene un DV 124 como se especifica en el 3D-HEVC actual, por ejemplo, usando la obtención NBDV, apuntando a una vista de referencia objetivo (V_0). A continuación, en la imagen

128 de la vista de referencia dentro de la misma unidad de acceso, el descodificador de vídeo 30 identifica el bloque de vídeo de referencia entrevista correspondiente 126 (**Base**) mediante DV 124.

2. El descodificador de vídeo 30 reutiliza la información de movimiento, por ejemplo, TMV 130, 132, del bloque de vídeo actual 120 para obtener la información de movimiento para el correspondiente bloque de vídeo de referencia entrevista 126. El descodificador de vídeo 30 puede aplicar una compensación de movimiento para el bloque de vídeo de referencia entrevista 126 en base a TMV 130, 132 del bloque de vídeo actual 120 y la imagen de referencia obtenida 142, 146 en la vista de referencia para el bloque de vídeo de referencia 126, para identificar un bloque de vídeo de referencia temporal compensado por movimiento 140, 144 (**BaseTRef**) y determinar el bloque predictor residual determinando **BaseTRef-Base**. La relación entre el bloque actual, el bloque correspondiente (**Base**) y el bloque compensado por movimiento (**BaseTRef**) se muestra en las FIGS. 9 y 10. En algunos ejemplos se selecciona como imagen de referencia del bloque correspondiente la imagen de referencia en la vista de referencia (V_0) que tiene el mismo valor POC (Recuento de Orden de Imágenes) que la imagen de referencia de la vista actual (V_m).

3. El descodificador de vídeo 30 puede aplicar un factor de ponderación w al bloque predictor residual para obtener un bloque predictor residual ponderado y añadir los valores del bloque residual ponderado a las muestras predichas para reconstruir el bloque de vídeo actual 120.

[0147] La FIG.11 es un diagrama conceptual de un ejemplo de estructura de predicción para ARP entrevista de un bloque de vídeo predicho de entrevista de acuerdo con las técnicas descritas en esta divulgación. Las técnicas relacionadas con la FIG. 11 fueron propuestos en Zhang et al., "CE4: Further improvements on advanced residual prediction", 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, 6ª Reunión: Ginebra, CH, 25 de octubre - 1 de noviembre de 2013, a continuación en el presente documento "JCT3V-F0123". JCT3V-F0123 se puede descargar desde el siguiente enlace: http://phenix.it-sudparis.eu/ict2/doc_end_user/current_document.php?id=1460.

[0148] De acuerdo con una técnica de ejemplo ilustrada en la FIG. 11, un codificador de vídeo, p. ej., codificador de vídeo 20 y/o descodificador de vídeo 30, puede usar un residual de entrevista de una unidad de acceso diferente para predecir el residual del bloque actual, que se predice mediante entrevista. En contraste con las propuestas de ARP en las que ARP no se realiza cuando el vector de movimiento del bloque actual es un DMV, y se realiza solo cuando el vector de movimiento para el bloque de vídeo actual es un TMV, la técnica de ejemplo de la FIG. 11 usa DMV para realizar ARP.

[0149] En particular, la técnica de ejemplo ilustrada en la FIG. 11 se puede realizar mediante un codificador de vídeo, por ejemplo, un codificador de vídeo 20 o un descodificador de vídeo 30, cuando el vector de movimiento del bloque de vídeo actual 150 (Curr) en una imagen actual 152 es un DMV 154 y el bloque de vídeo de referencia entrevista 156 (Base) en una imagen de referencia entrevista 158 en la vista de referencia (V_0) contiene al menos un TMV 160. En algunos ejemplos, el DMV 154 puede ser un DV que se convirtió a un DMV para actuar como un IDMVC para la predicción de información de movimiento del bloque de vídeo actual 150.

[0150] El codificador de vídeo identifica el bloque de vídeo de referencia entrevista 156 (**Base**) en la imagen de referencia entrevista 158 usando el DMV 154 para el bloque de vídeo actual 150. El codificador de vídeo usa un TMV 160 y una imagen de referencia asociada, por ejemplo, una imagen de referencia temporal 164 en la vista de referencia (V_0), del bloque de vídeo de referencia entrevista 156 junto con el DMV para identificar un bloque de vídeo de referencia temporal 162 (**BaseTRef**) en la imagen de referencia temporal 164 en la vista de referencia (V_0). La identificación del bloque de vídeo de referencia temporal 162 (**BaseTRef**) en base a TMV 160 y DMV 154 está representada por el vector de línea discontinua 170 (TMV + DMV). El codificador de vídeo también usa TMV 160 para identificar un bloque de vídeo de referencia temporal 166 (**CurrTRef**) en una imagen de referencia temporal 168 en la vista actual (V_m). El bloque de vídeo de referencia temporal 162 (**BaseTRef**) en la vista de referencia (V_0) y el bloque de vídeo de referencia temporal 166 (**CurrTRef**) en la vista actual (V_m) pueden estar dentro de la misma unidad de acceso, es decir, la imagen de referencia temporal 164 en la vista de referencia (V_0) y la imagen de referencia temporal 168 en la vista actual (V_m) pueden estar en la misma unidad de acceso.

[0151] El codificador de vídeo, por ejemplo, el codificador de vídeo 20 y/o el descodificador de vídeo 30, pueden a continuación calcular el bloque predictor residual de entrevista en una unidad de acceso diferente del bloque de vídeo actual 150 en base a la diferencia de píxel a píxel entre estos dos últimos bloques, es decir, la diferencia entre el bloque de vídeo de referencia temporal 166 en la vista actual y el bloque de vídeo de referencia temporal 162 en la vista de referencia, o **CurrTRef-BaseTRef**. La señal de diferencia, indicada como el predictor residual de entrevista, se puede usar para predecir el residual del bloque de vídeo actual 150. La señal de predicción del bloque de vídeo actual 150 puede ser la suma del predictor de entrevista, es decir, el bloque de vídeo de referencia entrevista 156 (**Base**), y el residual de entrevista predicho en una unidad de acceso diferente determinada en base a la diferencia entre el bloque de vídeo de referencia temporal 166 en la vista actual y el bloque de vídeo de referencia temporal 162 en la vista de referencia. En algunos ejemplos, se aplica un factor de ponderación w al residual de entrevista predicho en la unidad de acceso diferente. En dichos ejemplos, la señal de predicción del bloque de vídeo actual 150 puede ser: **Base + w*(CurrTRef-BaseTRef)**.

[0152] En algunos ejemplos, el codificador de vídeo puede determinar una imagen de referencia objetivo en una unidad de acceso objetivo para ARP entrevista, por ejemplo, similar a la determinación de una imagen de referencia objetivo para ARP temporal, como se analizó anteriormente. En algunos ejemplos, como se ha analizado anteriormente con referencia a JCT3V-D0177, la imagen de referencia objetivo para cada lista de imágenes de referencia es la primera imagen de referencia en la lista de imágenes de referencia. En otros ejemplos, la imagen de referencia objetivo, por ejemplo, POC objetivo, para una o ambas listas de imágenes de referencia puede ser señalizada desde el codificador de vídeo 20 al descodificador de vídeo 30, por ejemplo, en una PU, CU, fragmento, imagen u otra base. En otros ejemplos, la imagen de referencia objetivo para cada lista de imágenes de referencia es la imagen de referencia temporal en la lista de imágenes de referencia que tiene la diferencia de POC más pequeña comparada con el bloque actual y el índice de imagen de referencia más pequeño. En otros ejemplos, la imagen de referencia objetivo para dos listas de imágenes de referencia es la misma.

[0153] Si la imagen que contiene el bloque de vídeo de referencia temporal en la vista de referencia indicada por TMV 160 está en una unidad de acceso diferente (instancia de tiempo) a la imagen de referencia de ARP objetivo, el codificador de vídeo puede escalar TMV 160 a la imagen de referencia objetivo, por ejemplo, imagen de referencia objetivo 164, para identificar el bloque de vídeo de referencia temporal 162 (**BaseTRef**) en la vista de referencia para ARP entrevista. En tales ejemplos, el codificador de vídeo localiza el bloque de vídeo de referencia temporal 162 en la unidad de acceso que contiene la imagen de referencia de ARP objetivo. El codificador de vídeo puede escalar TMV 160 mediante escalado de POC. Además, el TMV escalado se usa para identificar el bloque de vídeo de referencia temporal (**CurrTRef**) 166 en la vista actual que se encuentra en la imagen de referencia de ARP objetivo.

[0154] En algunos ejemplos, el codificador de vídeo escala TMV 160 a la imagen de referencia objetivo LX (X siendo 0 o 1), donde LX corresponde a la RefPicListX de la PU que incluye el TMV. En algunos ejemplos, el codificador de vídeo puede escalar TMVs desde uno o ambos de RefPicList0 o RefPicList1 a la imagen de referencia objetivo L0 o L1, respectivamente. En algunos ejemplos, el codificador de vídeo escala TMV 160 a la imagen de referencia objetivo LX, en la que X satisface la condición de que el DMV 154 del bloque de vídeo actual 150, por ejemplo, la PU actual, corresponde a RefPicListX.

[0155] De forma similar, en algunos ejemplos, el codificador de vídeo escala DMV 154 a una vista de referencia objetivo para ARP antes de identificar el bloque de vídeo de referencia entrevista 156 en la imagen de referencia 158 en la vista de referencia objetivo. El codificador de vídeo puede escalar el DMV 154 mediante escalado de diferencia de orden de vista. La vista de referencia objetivo puede ser predeterminada y conocida por el codificador de vídeo 20 y el descodificador de vídeo 30, o puede ser señalizada desde el codificador de vídeo 20 al descodificador de vídeo 30, por ejemplo, en una PU, CU, fragmento, imagen u otra base.

[0156] En algunos ejemplos de ARP entrevista, el codificador de vídeo, por ejemplo, el codificador de vídeo 20 y/o el descodificador de vídeo 30, puede derivar la señal de predicción para el bloque actual 150 usando la misma estructura de predicción y los bloques de vídeo de referencia identificados 156, 162 y 166 ilustrados en la FIG. 11, pero determinar el bloque predictor residual en base a la diferencia entre los bloques de referencia 156 y 162 en la vista de referencia, en lugar de los bloques de referencia 162 y 166 en la unidad de acceso diferente. En tales ejemplos, el codificador de vídeo puede aplicar un factor de ponderación a otras matrices de muestra, por ejemplo, la diferencia entre los bloques de referencia 156 y 162 en la vista de referencia y, en consecuencia, obtener la señal de predicción para el bloque de vídeo actual 150 de la forma siguiente: **CurrTRef** + $w \cdot (\text{Base} - \text{BaseTRef})$. En algunos ejemplos de ARP entrevista, el codificador de vídeo puede usar diversos filtros de interpolación, incluyendo filtros bilineales, para obtener los bloques de vídeo de referencia 156, 162 y 166 si están alineados con posiciones de píxeles fraccionadas.

[0157] Aunque la FIG. 11 ilustra ejemplos de ARP entrevista en los que se identifican los bloques de vídeo de referencia temporales en las vistas actuales y de referencia usando el TMV asociado con el bloque de referencia entrevista y la imagen de referencia asociada del bloque de vídeo de referencia entrevista, en otros ejemplos se pueden usar otras TMVs e imágenes de referencia asociadas para identificar los bloques de vídeo de referencia temporales en las vistas actual y de referencia. Por ejemplo, si el DMV del bloque de vídeo actual es de una primera lista de imágenes de referencia (por ejemplo, RefPicList0 o RefPicList1) del bloque de vídeo actual, un codificador de vídeo puede usar un TMV correspondiente a la segunda lista de imágenes de referencia del bloque actual y una imagen de referencia asociada de una segunda lista de imágenes de referencia (por ejemplo, la otra de RefPicList0 o RefPicList1) del bloque de vídeo actual. En dichos ejemplos, el codificador de vídeo puede identificar el bloque de vídeo de referencia temporal en la vista actual en la imagen de referencia asociada con el TMV, o escalar el TMV a una unidad de acceso objetivo y una imagen de referencia objetivo para ARP para identificar el bloque de vídeo de referencia temporal en la vista actual. En tales ejemplos, el codificador de vídeo puede identificar el bloque de vídeo de referencia temporal en la vista de referencia en una imagen de referencia que está en la misma unidad de acceso que la imagen de referencia en la que se localizó el bloque de vídeo de referencia temporal en la vista actual. En otros ejemplos, en lugar del TMV del bloque de vídeo de referencia entrevista o el TMV de la otra lista de imágenes de referencia del bloque de vídeo actual, un codificador de vídeo puede usar de manera similar un TMV y una imagen de referencia asociada derivada de la información de movimiento de los bloques de vídeo contiguos espaciales o temporales del bloque de vídeo actual para identificar los bloques de vídeo de referencia temporales en las vistas actuales y de referencia para ARP.

[0158] En la siguiente descripción, si la referencia correspondiente para una lista de imágenes de referencia es una imagen de referencia temporal y se aplica ARP, el proceso ARP se indica como ARP temporal. De otro modo, si la referencia correspondiente para una lista de imágenes de referencia es una imagen de referencia entrevista y se aplica ARP, el proceso ARP se indica como ARP entrevista.

[0159] En algunas propuestas de ARP, se pueden usar tres factores de ponderación, es decir, 0, 0,5 y 1. El factor de ponderación que da lugar a coste de tasa de distorsión mínimo para la CU actual se selecciona como el factor de ponderación final y se transmite el índice del factor de ponderación correspondiente (0, 1 y 2 que corresponden al factor de ponderación 0, 1 y 0,5, respectivamente) en el flujo de bits a nivel de CU. Todas las predicciones de PU en una CU comparten el mismo factor de ponderación. Cuando el factor de ponderación es igual a 0, la ARP no se usa para la CU actual.

[0160] Los aspectos adicionales relacionados con ARP para 3D-HEVC se describen en Zhang y col., "3D-CE4: Advanced residual prediction for multiview coding", Equipo de Colaboración Conjunta en el Desarrollo de la Extensión de Codificación de Vídeo de ITU-T SG 16 WP 3 e ISO/IEC JTC1/SC 29/WG 11, 3.ª Reunión: Ginebra, Suiza, 17-23 de enero de 2013, Doc. JCT3V-C0049 (número MPEG m27784), a continuación en el presente documento "JCT3V-C0049". JCT3V-C0049 se puede descargar desde el siguiente enlace: http://phenix.int-evry.fr/ict3v/doc_end_user/current_document.php?id=487_.

[0161] En JCT3V-C0049, las imágenes de referencia de diferentes PU codificadas con factores de ponderación distintos a cero pueden ser diferentes de PU a PU (o de bloque de vídeo actual a bloque de vídeo actual). Por lo tanto, puede ser necesario acceder a imágenes diferentes de la vista de referencia para generar el bloque con compensación de movimiento (BaseTRef), por ejemplo, bloques de vídeo de referencia temporal 116, 140 y 144 en las FIGS. 9 y 10, o el correspondiente bloque de vídeo de referencia entrevista en la vista de referencia (Base), por ejemplo, los bloques de vídeo de referencia entrevista 106 y 126 en las FIGS. 9 y 10.

[0162] Cuando el factor de ponderación no es igual a 0, para un residual temporal, los vectores de movimiento de la PU actual se ajustan a escala hacia una imagen fija antes de realizar la compensación de movimiento para ambos procesos de generación residual y de predictor residual. Cuando se aplica ARP a un residual entrevista, los vectores de movimiento temporal del bloque de referencia (es decir, el bloque 156 en la FIG. 11) se ajustan a escala hacia una imagen fija antes de realizar la compensación de movimiento para ambos procesos de generación residual y de predictor residual. Para ambos casos (es decir, un residual temporal o residual entrevista), la imagen fija se define como la primera imagen de referencia temporal disponible de cada lista de imágenes de referencia. Cuando el vector de movimiento descodificado no apunta a la imagen fija, primero se ajusta a escala y a continuación se usa para identificar CurrTRef y BaseTRef.

[0163] Dicha imagen de referencia usada para ARP se denomina imagen de referencia de ARP objetivo. Téngase en cuenta que cuando el fragmento actual es un fragmento B, la imagen de referencia de ARP objetivo está asociada con una lista de imágenes de referencia particular. Por lo tanto, se pueden usar dos imágenes de referencia de ARP objetivo.

[0164] Se puede realizar una verificación de disponibilidad de las imágenes de referencia ARP objetivo como sigue. En primer lugar, indicar la imagen de referencia de ARP objetivo asociada a una lista de imágenes de referencia X (X siendo igual a 0 o 1) por RpRefPicLX, e indicar la imagen en la vista con índice de orden de vista igual al obtenido del proceso de obtención NBDV, y con el mismo valor POC de RpRefPicLX por RefPicInRefViewLX.

[0165] Cuando una de las siguientes condiciones es falsa, ARP se deshabilita para la lista de imágenes de referencia X:

- RpRefPicLX no está disponible
- RefPicInRefViewLX no está almacenada en la memoria intermedia de imágenes descodificadas
- RefPicInRefViewLX no está incluido en ninguna de las listas de imágenes de referencia del bloque correspondiente (por ejemplo, el bloque 106 en la FIG. 9 o el bloque 156 en la FIG. 11) localizado por el DV del proceso de obtención de NBDV o el DMV asociado con el bloque actual, ARP se deshabilita para esta lista de imágenes de referencia.

[0166] Cuando se aplica ARP, se puede usar un filtro bilineal al generar el residual y predictor residual. Es decir, por ejemplo, los bloques 106, 112 y 116 de la FIG. 9 se generan usando un filtro bilineal.

[0167] La FIG. 12 es un diagrama conceptual que ilustra una estructura de predicción de ejemplo para ARP bidireccional usando predicción entrevista para una lista de imágenes de referencia y predicción temporal para otra lista de imágenes de referencia. La técnica de ejemplo de la FIG. 12 se puede realizar mediante el codificador de vídeo 20 y/o el descodificador de vídeo 30 cuando una dirección de predicción para la predicción bidireccional del bloque de vídeo actual 250 usa la predicción temporal (por ejemplo, para la lista de imágenes de referencia X), y la otra dirección

de predicción del bloque de vídeo actual 250 usa predicción entrevista (por ejemplo, para la lista de imágenes de referencia Y ($Y=1-X$)).

[0168] En el ejemplo de la FIG. 12, el bloque de vídeo actual 250 se puede asociar con un TMV 210 y un DMV 254. El codificador de vídeo 20 y/o el decodificador de vídeo 30 se pueden configurar para identificar y acceder a bloques de referencia para la lista de imágenes de referencia X (es decir, una primera dirección de predicción) de una manera similar a la descrita anteriormente con referencia a la FIG. 9.

[0169] El codificador de vídeo 20 y/o el decodificador de vídeo 30 identifica los siguientes bloques en la predicción del residual para un bloque de vídeo actual 250 en una imagen actual 253 de la vista actual V_m , por ejemplo, dependiente. El codificador de vídeo 20 y/o el decodificador de vídeo 30 puede identificar un bloque de vídeo de referencia entrevista 206 (**BaseX**) en una imagen de referencia entrevista 258 de una vista de referencia/base (V_0 en la FIG. 12). El codificador de vídeo 20 y/o el decodificador de vídeo 30 identifican el bloque de vídeo de referencia entrevista 206 basado en DV 204 del bloque de vídeo actual 250 (**Curr**). El codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden determinar DV 204 usando la obtención NBDV, como se describió anteriormente.

[0170] El codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden identificar además un bloque de vídeo de referencia temporal 212 (**CurrTRefX**) en una imagen de referencia temporal 270 en la misma vista (V_m) como el bloque de vídeo actual 250 (**Curr**). El codificador de vídeo 20 y/o el decodificador de vídeo 30 identifican el bloque de vídeo de referencia temporal 212 usando TMV 210 del bloque de vídeo actual 250. El codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden determinar TMV 210 usando cualquiera de las técnicas descritas en el presente documento. El codificador de vídeo 20 y/o el decodificador de vídeo 30 puede identificar además un bloque de vídeo de referencia temporal 216 (**BaseTRefX**) en una imagen de referencia temporal 272 en la vista de referencia, es decir, la misma vista que el bloque de vídeo de referencia entrevista 206 (**BaseX**). El codificador de vídeo 20 y/o el decodificador de vídeo 30 puede identificar el bloque de vídeo de referencia temporal 216 en la vista de referencia usando TMV 210 del bloque de vídeo actual 250 (**Curr**). Un vector 220 de TMV 210 + DV 204 puede identificar el bloque de vídeo de referencia temporal 216 (**BaseTRefX**) con respecto al bloque de vídeo actual 250 (**Curr**). Como se puede ver en la FIG. 12, para la lista de imágenes de referencia X (es decir, una primera dirección de predicción), el codificador de vídeo 20 y/o el decodificador de vídeo 30 están configurados para identificar y acceder a tres bloques de referencia (es decir, bloques de referencia 206, 212 y 216).

[0171] El codificador de vídeo 20 y/o el decodificador de vídeo 30 se pueden configurar para identificar y acceder a bloques de referencia para la lista de imágenes de referencia Y (es decir, una primera dirección de predicción) de una manera similar a la descrita anteriormente con referencia a la FIG. 11. El codificador de vídeo 20 y/o el decodificador de vídeo 30 puede identificar un bloque de vídeo de referencia entrevista 256 (**BaseY**) en una imagen de referencia entrevista 258 de una vista de referencia/base (V_0 en la FIG. 12). El codificador de vídeo 20 y/o el decodificador de vídeo 30 identifican el bloque de vídeo de referencia entrevista 256 basado en DMV 254 del bloque de vídeo actual 250 (**Curr**).

[0172] El codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden identificar además un bloque de vídeo de referencia temporal 273 (**CurrTRefY**) en una imagen de referencia temporal 268 en la misma vista (V_m) como el bloque de vídeo actual 250 (**Curr**). El codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden identificar el bloque de vídeo de referencia temporal 273 usando TMV' 285 del bloque de vídeo actual 250. El codificador de vídeo usa un TMV' 285 y una imagen de referencia asociada, por ejemplo, una imagen de referencia temporal 265 en la vista de referencia (V_0), del bloque de vídeo de referencia entrevista 256 junto con el DMV 254 para identificar un bloque de vídeo de referencia temporal 271 (**BaseTRefY**) en la imagen de referencia temporal 265 en la vista de referencia (V_0). La identificación del bloque de vídeo de referencia temporal 271 (**BaseTRefY**) en base a TMV' 285 y DMV 254 está representada por el vector de línea discontinua 170 (TMV' + DMV). El bloque de vídeo de referencia temporal 271 (**BaseTRefY**) en la vista de referencia (V_0) y el bloque de vídeo de referencia temporal 273 (**CurrTRefY**) en la vista actual (V_m) puede estar dentro de la misma unidad de acceso, es decir, la imagen de referencia temporal 265 en la vista de referencia (V_0) y la imagen de referencia temporal 268 en la vista actual (V_m) pueden estar en la misma unidad de acceso.

[0173] Como se puede ver en la FIG. 12, para la lista de imágenes de referencia Y (es decir, una segunda dirección de predicción), el codificador de vídeo 20 y/o el decodificador de vídeo 30 están configurados para identificar y acceder a tres bloques de referencia adicionales (es decir, bloques de referencia 256, 271 y 273).

[0174] Las técnicas mencionadas anteriormente para ARP en 3D-HEVC presentan varias desventajas. Como un ejemplo, el número de accesos a memoria para la información de movimiento incrementa cuando se realiza ARP a nivel de bloque o ARP a nivel de PU junto con la predicción bidireccional, ya que la predicción bidireccional incluye inherentemente el uso de información de movimiento para dos listas de imágenes de referencia diferentes. Además, el número de bloques de referencia que se identifican y acceden es elevado. Por tanto, la predicción bidireccional en combinación con ARP incrementa la complejidad del decodificador.

[0175] Esta divulgación propone diversas técnicas de ejemplo para abordar el problema mencionado anteriormente para ARP para reducir la complejidad del decodificador de vídeo. Cada una de las técnicas enumeradas a

continuación disminuye el número de accesos a memoria necesarios para realizar ARP y otras técnicas de codificación de vídeo asociadas en relación con las propuestas actuales para ARP.

[0176] La FIG. 13 es un diagrama conceptual que ilustra una estructura de predicción de ejemplo para ARP bidireccional usando predicción entrevista para una lista de imágenes de referencia y predicción temporal para otra lista de imágenes de referencia de acuerdo con las técnicas de esta divulgación. En el ejemplo de la FIG. 13, el codificador de vídeo 20 y/o el descodificador de vídeo 30 están configurados para codificar el bloque de vídeo actual 250 usando predicción bidireccional y ARP. La predicción bidireccional incluye predicción temporal para la lista de imágenes de referencia X (por ejemplo, una primera dirección de predicción) y predicción entrevista para la lista de imágenes de referencia Y (por ejemplo, una segunda dirección de predicción).

[0177] De acuerdo con las técnicas de la FIG. 13, el codificador de vídeo 20 y/o el descodificador de vídeo 30 están configurados para identificar el bloque de referencia 206 (**BaseX**), el bloque de referencia 216 (**BaseTrefX**) y el bloque de referencia 212 (**CurrTrefX**) para la lista de imágenes de referencia X (por ejemplo, una primera dirección de predicción) de la misma manera que se describe anteriormente con referencia a la FIG. 12. Es decir, TMV 210 se usa para identificar el bloque de referencia 216 (**BaseTrefX**) y el bloque de referencia 212 (**CurrTrefX**) relativo al bloque de referencia 206 (**BaseX**) y el bloque de vídeo actual 250, respectivamente. Además, el codificador de vídeo 20 y/o el descodificador de vídeo 30 están configurados para identificar el bloque de referencia 256 (**BaseY**) para la lista de imágenes de referencia Y (por ejemplo, una segunda dirección de predicción) usando DMV 254 (es decir, de la misma manera que se describe anteriormente con referencia a la FIG. 12).

[0178] Sin embargo, el codificador de vídeo 20 y/o el descodificador de vídeo 30 no usan información de movimiento temporal asociada con el bloque de referencia 256 (**BaseY**) para identificar **CurrTrefY** y **BaseTrefY**. En cambio, de acuerdo con las técnicas de esta divulgación, el codificador de vídeo 20 y/o el descodificador de vídeo 30 se pueden configurar para usar la información de movimiento temporal para la lista de referencia X (es decir, TMV 210) para identificar **CurrTrefY** y **BaseTrefY**. Como se muestra en la FIG. 13, el codificador de vídeo 20 y/o el descodificador de vídeo 30 están configurados para identificar el bloque de referencia 290 (**BaseTrefY**) en la vista V_0 usando TMV 210 relativo al bloque de referencia 256 (**BaseY**). Es decir, el codificador de vídeo 20 y/o el descodificador de vídeo 30 están configurados para identificar el bloque de referencia 290 (**BaseTrefY**) usando DMV 254 y TMV 210. El codificador de vídeo 20 y/o el descodificador de vídeo 30 se configuran además para identificar **CurrTrefY** en la misma vista (V_m) que el bloque de vídeo actual 250 usando TMV 210. Como tal, el bloque de referencia 212 sirve como **CurrTrefX** y **CurrTrefY**. Por tanto, usando las técnicas de esta divulgación, el codificador de vídeo 20 y/o el descodificador de vídeo 30 solo identifican y acceden a 5 bloques de referencia, en lugar de 6, cuando se realiza ARP con predicción bidireccional.

[0179] En resumen, la identificación de bloques de referencia para ARP entrevista correspondientes a la lista de imágenes de referencia Y (por ejemplo, una segunda dirección de predicción), el codificador de vídeo 20 y el descodificador de vídeo 30 pueden usar la información de movimiento temporal (por ejemplo, TMV 210 en la FIG. 13) asociada con predicción temporal para la lista de imágenes de referencia X para identificar los bloques de referencia en una unidad de acceso diferente (es decir, bloques de referencia 290 y 212). En este caso, al realizar la ARP entrevista, no es necesario generar el bloque de referencia en una unidad de acceso diferente a la vista actual (es decir, el bloque de referencia 212), ya que es el mismo que el bloque de referencia identificado para ARP temporal para la lista de imágenes de referencia X. Es decir, el bloque de referencia 212 se usa tanto para ARP temporal como para ARP entrevista.

[0180] De esta manera, la información de movimiento temporal para una primera dirección de predicción se reutiliza para una segunda dirección de predicción. Como tal, se deben realizar menos accesos a memoria de la información de movimiento temporal, ya que no es necesario acceder a la información de movimiento temporal del bloque identificado por el vector de movimiento del primer bloque codificado correspondiente a la segunda dirección de predicción, por tanto permite una descodificación de vídeo más rápida. Además, el número total de bloques de referencia usados al realizar ARP se puede reducir de 6 a 5, lo que resulta en una menor complejidad computacional en términos de interpolación usando funcionamientos de multiplicación y suma. Asimismo, cuando se realiza la interpredicción bidireccional, el codificador de vídeo 20 se puede configurar para reutilizar la información de movimiento temporal para una primera dirección de predicción cuando se codifica la segunda dirección de predicción.

[0181] En otro ejemplo de la divulgación, el codificador de vídeo 20 y el descodificador de vídeo 30 se pueden configurar para realizar un proceso ARP simplificado cuando una dirección de predicción (por ejemplo, correspondiente a la lista de imágenes de referencia X) de la predicción bidireccional corresponde a una imagen de referencia temporal y la otra dirección de predicción (por ejemplo, correspondiente a la lista de imágenes de referencia Y) corresponde a una imagen de referencia entrevista. En este caso, para ARP temporal correspondiente a la lista de imágenes de referencia X, el codificador de vídeo 20 y el descodificador de vídeo 30 se pueden configurar para usar el vector de movimiento de disparidad (MVY) asociado con la imagen de referencia entrevista para identificar los bloques de referencia en la vista de referencia (por ejemplo, el bloque de referencia 273 en la FIG. 12) en lugar de usar un vector de disparidad derivado de un proceso de obtención de NBDV/DoNBDV. Mientras tanto, el vector de disparidad del proceso NBDV o DoNBDV se mantiene cambiado, lo que aún se puede usar en la predicción de movimiento entrevista para generar el IPMVC o IDMVC.

[0182] Cabe señalar que los procedimientos anteriores se pueden aplicar tanto a ARP a nivel de PU como a ARP a nivel de bloque. ARP a nivel de PU y a nivel de bloque se describirá con más detalle a continuación.

[0183] Ahora se analizarán técnicas para ARP a nivel de bloque. A diferencia de la descripción anterior, en la que todos los bloques dentro de una PU comparten la misma información de movimiento para ARP (a veces llamado ARP a nivel de PU), en ARP a nivel de bloque, una PU se divide en varios subbloques (por ejemplo, subbloques 8x8) y cada subbloque está asociado con su propia información de movimiento derivada para realizar ARP. Es decir, cada subbloque comparte la misma información de movimiento que la PU actual. Sin embargo, el vector de movimiento obtenido (es decir, un vector de disparidad en ARP temporal o un vector de movimiento temporal en ARP entrevista) se puede actualizar para cada subbloque.

[0184] La FIG. 14 es un diagrama conceptual que ilustra ARP temporal basada en bloques. Como se muestra en la FIG. 14, la imagen actual 302 incluye un bloque actual 300 (Curr) dividido en cuatro subbloques 300a, 300b, 300c y 300d. El vector de movimiento 310 (mvLX) es el vector de movimiento usado para realizar la interpolación en el bloque actual 300. El vector de movimiento 310 apunta a un bloque de referencia 312 (CurrTref), que incluye subbloques 312a-d, en la imagen de referencia 314. La imagen actual 302 y la imagen de referencia 314 están en la misma vista (Vm).

[0185] Para ARP temporal basado en bloques, se usa un vector de movimiento derivado predeterminado para cada uno de los subbloques 300a-d. Para ARP temporal, el vector de movimiento derivado predeterminado es un vector de disparidad indicado por DV[i] para el subbloque i-ésimo en la FIG. 14, y se puede derivar usando un proceso de obtención NBDV, como en ARP actual. Es decir, se puede realizar un proceso de obtención NBDV para cada uno de los subbloques 300a-d para derivar un DV para cada uno de los subbloques 300a-d. Cada uno de los DV derivados apunta a un bloque de referencia particular 306a-d (Base) en la vista de referencia 308. Por ejemplo, DV 304 (DV[0]) apunta al bloque de referencia 306a y DV 305 (DV[1]) apunta al bloque de referencia 306b.

[0186] La vista de referencia 308 está en la misma instancia de tiempo que la imagen actual 302, pero está en otra vista. Cuando la posición central de uno de los subbloques 312a-d dentro del bloque de referencia 312 contiene un vector de movimiento de disparidad, el vector de disparidad DV[i] para el correspondiente del subbloque actual 300a-d se actualiza para usar ese vector de movimiento de disparidad. Es decir, por ejemplo, si la posición central del subbloque de referencia 312a, correspondiente al subbloque actual 300a, tiene un vector de movimiento de disparidad asociado, el vector de movimiento de disparidad asociado con el subbloque de referencia 312a se usa como vector de disparidad para el subbloque 300a.

[0187] Una vez que se ha identificado cada uno de los bloques de referencia 306a-d, el vector de movimiento 310 se puede usar para encontrar los bloques de referencia 316a-d (BaseTref) en la imagen de referencia 318. La imagen de referencia 318 está en una instancia de tiempo diferente, así como una vista diferente, que la imagen actual 302. Los predictores residuales se pueden a continuación determinar restando los bloques de referencia 306a-d (Base) de los correspondientes bloques de referencia 316a-d (BaseTref). A continuación, se puede realizar ARP para cada uno de los subbloques 300a-d.

[0188] La FIG. 15 es un diagrama conceptual que ilustra ARP entrevista basada en bloques. Como se muestra en la FIG. 15, la imagen actual 352 incluye un bloque actual 350 (Curr) dividido en cuatro subbloques 350, 350b, 350c y 350d. El vector de movimiento de disparidad 360 (DMV) es el vector de movimiento de disparidad usado para realizar la predicción entrevista en el bloque 350 actual. El vector de movimiento de disparidad 360 apunta a un bloque de referencia 356 (Base), que incluye subbloques 356a-d, en la imagen de referencia 358. La imagen actual 352 y la imagen de referencia 358 están en la misma instancia de tiempo, pero en diferentes vistas.

[0189] Para ARP entrevista basado en bloques, se usa un vector de movimiento derivado predeterminado para cada uno de los subbloques 350a-d. Para ARP entrevista, el vector de movimiento predeterminado por defecto es un vector de movimiento indicado por mvLX[i] para el subbloque i-ésimo en la FIG. 15, y se puede establecer como el vector de movimiento temporal que cubre la posición central de cada uno de los subbloques 356a-d, como en ARP actual. Cuando el bloque que cubre la posición central del i-ésimo bloque 8x8 dentro de los subbloques 356 contiene un vector de movimiento temporal, mvLX[i] se actualiza para que sea ese vector de movimiento temporal.

[0190] Cada uno de los vectores de movimiento derivados apunta a un bloque de referencia particular 366a-d (BaseTref) en la vista de referencia 368. Por ejemplo, el vector de movimiento 354 (mvLX[0]) apunta al bloque de referencia 368a y el vector de movimiento 355 (mvLX[3]) apunta al bloque de referencia 366d.

[0191] Una vez que se ha identificado cada uno de los bloques de referencia 366a-d, se puede usar el vector de movimiento de disparidad 360 para encontrar los bloques de referencia 362a-d (CurrTref) en la imagen de referencia 364. La imagen de referencia 364 está en una instancia de tiempo diferente que la imagen actual 352. Los predictores residuales se pueden a continuación determinar restando los bloques de referencia 362a-d (CurrTref) de los correspondientes bloques de referencia 366a-d (BaseTref). A continuación, se puede realizar ARP para cada uno de los subbloques 350a-d.

[0192] Como se describió anteriormente, para ARP temporal basado en bloques, se accede al vector de movimiento 310 y se usa para localizar el bloque de referencia 312 (CurrTref). Del mismo modo, para ARP entrevista basada en bloques, se accede al vector de movimiento de disparidad 360 y se usa para localizar el bloque de referencia 356 (Base).

[0193] La FIG. 16 es un diagrama conceptual que ilustra ARP basada en bloques con un candidato de fusión sub-PU. Cuando se habilita la predicción de movimiento entrevista de sub-PU, se accede a la información de movimiento de un bloque de referencia (406) identificado por el vector de disparidad 410 derivado de un proceso de obtención NBDV/DoNBDV para derivar el candidato de fusión sub-PU. Después de que se determina el candidato de fusión sub-PU, es decir, para cada sub-PU dentro del bloque 400 (Curr), tendrá su información de movimiento temporal, como se indica por el vector de movimiento 404 (mvLX[0]) y el vector de movimiento 405 (mvLX[1]) como se muestra en la FIG. 14. Los vectores de movimiento 404 y 405 se pueden usar para identificar los bloques de referencia 412 (CurrTref) y los bloques de referencia 416 (BaseTref).

[0194] Cuando se invoca un proceso ARP, también se accede a la información de movimiento de cada subbloque (por ejemplo, bloque 8x8) dentro del bloque de referencia 412 (CurrTref). Cuando el correspondiente subbloque 412a-d (CurrRef) está asociado con un vector de movimiento de disparidad, el vector de movimiento de disparidad se puede usar para localizar un bloque de referencia en una vista de referencia (por ejemplo, el bloque 406).

[0195] Por lo tanto, puede ser necesario acceder a la información de movimiento de dos bloques. Es decir, se accede a la información de movimiento para un bloque identificado por un DV de un proceso NBDV/DoNBDV para el candidato de fusión entrevista de sub-PU. Además, la información de movimiento es el acceso para un bloque identificado por cualquier información de movimiento temporal derivada.

[0196] Las técnicas mencionadas anteriormente para ARP en 3D-HEVC presentan varias desventajas. Como un ejemplo, cuando se usan tanto la predicción de fusión entrevista de sub-PU como ARP temporal a nivel de bloque para codificar una PU, se accede a la información de movimiento de dos bloques de referencia. Uno es un bloque de referencia en una vista de referencia que se identifica mediante un vector de disparidad derivado de un proceso de obtención DoNBDV/NBDV. Además, se accede a la información de movimiento correspondiente para derivar el candidato de fusión entrevista de sub-PU. Después de derivar el candidato de fusión de entrevista de sub-PU, se accede a otro bloque en una imagen de referencia temporal para comprobar si el bloque en la imagen de referencia temporal contiene un vector de movimiento de disparidad. El doble acceso a la información de movimiento asociada con diferentes bloques incrementa significativamente la complejidad del diseño del descodificador de vídeo y puede disminuir el rendimiento del descodificador.

[0197] Como otra desventaja, cuando se usa ARP sub-PU (es decir, a nivel de bloque), los vectores de movimiento de disparidad asociados con el bloque de referencia apuntado por el vector de movimiento temporal de un bloque actual se usan para actualizar el vector de disparidad predeterminado. Para un bloque de subbloque, incluso si el subbloque tiene el mismo vector de movimiento de disparidad que sus bloques contiguos (ya sea a la izquierda, arriba, abajo o derecha), el proceso ARP todavía se realiza en cada subbloque, incrementando por tanto la complejidad del descodificador de vídeo.

[0198] Esta divulgación propone diversas técnicas de ejemplo para abordar el problema mencionado anteriormente para ARP para reducir la complejidad del descodificador de vídeo. Cada una de las técnicas enumeradas a continuación disminuye el número de accesos a memoria necesarios para realizar ARP y otras técnicas de codificación de vídeo asociadas en relación con las propuestas actuales para ARP.

[0199] En un ejemplo de la divulgación, cuando se habilita la predicción de movimiento entrevista de sub-PU y se aplica un candidato de fusión de entrevista de sub-PU (que corresponde a la información de movimiento temporal) a la PU actual, el codificador de vídeo 20 y el descodificador de vídeo 30 se pueden configurar para deshabilitar ARP a nivel de bloque. En su lugar, se puede habilitar ARP a nivel de PU.

[0200] Cuando se habilita la predicción de movimiento entrevista de sub-PU, y se aplica ARP a nivel de PU, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden determinar la información de movimiento temporal para cada sub-PU. Es decir, cada sub-PU tiene su propia información de movimiento temporal. Sin embargo, el codificador de vídeo 20 y el descodificador de vídeo 30 determinan el mismo vector de disparidad para todas las sub-PU. La información de movimiento temporal y el vector de disparidad se usan para obtener el residual y el predictor residual, como se describió anteriormente. Téngase en cuenta que cuando se aplica la predicción de movimiento entrevista de sub-PU, el proceso ARP usado es ARP temporal.

[0201] Las siguientes técnicas de ejemplo se usan cuando no se usa la predicción de movimiento entrevista de sub-PU. En un ejemplo, para la ARP temporal, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden determinar un vector de disparidad para cada sub-PU. En un ejemplo, el vector de disparidad determinado puede ser información de movimiento de disparidad derivada del bloque de referencia de la sub-PU actual identificada por la información de movimiento temporal de la sub-PU actual en la imagen de referencia temporal. Para ARP entrevista, el codificador de

vídeo 20 y el decodificador de vídeo 30 pueden determinar información de movimiento temporal para cada sub-PU. En un ejemplo, la información de movimiento temporal se puede derivar del bloque de referencia de la sub-PU actual identificada por la información de movimiento de disparidad de la sub-PU actual en la imagen de referencia entrevista.

[0202] En otro ejemplo de la divulgación, cuando se habilita la predicción de movimiento entrevista de sub-PU, el codificador de vídeo 20 y el decodificador de vídeo se pueden configurar para deshabilitar ARP a nivel de bloque para una dirección de predicción correspondiente a una lista de imágenes de referencia particular si la imagen de referencia asociada es una imagen de referencia temporal. En este caso, el codificador de vídeo 20 y el decodificador de vídeo 30 se pueden configurar para habilitar únicamente ARP a nivel de PU para esta dirección de predicción.

[0203] En un ejemplo, se aplica el siguiente proceso: Si la PU actual usa un candidato de fusión entrevista, el codificador de vídeo 20 y el decodificador de vídeo 30 determinan la información de movimiento temporal para cada sub-PU. Sin embargo, el codificador de vídeo 20 y el decodificador de vídeo 30 determinan el mismo vector de disparidad para todas las sub-PU. La información de movimiento temporal y el vector de disparidad se usan para obtener el residual y el predictor residual, como se describió anteriormente.

[0204] De otro modo, si la PU actual usa uno de los otros candidatos de fusión disponibles (es decir, no un candidato de fusión entrevista), el codificador de vídeo 20 y el decodificador de vídeo 30 aplican ARP temporal a nivel de PU, en el que todos los bloques dentro de la PU actual comparten la misma información de movimiento, para una dirección de predicción si la imagen de referencia correspondiente es una imagen de referencia temporal. Para la una dirección de predicción, si la imagen de referencia correspondiente es una imagen de referencia entrevista, el codificador de vídeo 20 y el decodificador de vídeo 30 usan ARP entrevista a nivel de PU, en el que todos los bloques dentro de la PU actual comparten la misma información de movimiento. En este caso, también se puede aplicar ARP a nivel de bloque en el que los bloques dentro de la PU actual pueden compartir la misma información de movimiento de disparidad y diferente información de movimiento temporal.

[0205] En otro ejemplo de la divulgación, cuando se habilita ARP a nivel de bloque, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden determinar el tamaño de bloque para realizar ARP en base a la información de movimiento. En un ejemplo, para una dirección de predicción correspondiente a una lista de imágenes de referencia particular, si la imagen de referencia correspondiente es una imagen de referencia temporal, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden usar ARP a nivel de bloque. En este caso, un bloque actual tiene la misma información de movimiento de disparidad que su bloque contiguo (por ejemplo, bloques contiguos izquierdo, superior, inferior y/o derecho). Además, el bloque actual y los bloques contiguos se fusionan y ARP se realiza una vez para los bloques fusionados.

[0206] En otro ejemplo de la divulgación, para una dirección de predicción correspondiente a una lista de imágenes de referencia, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden usar ARP a nivel de bloque si la imagen de referencia correspondiente es una imagen de referencia entrevista. En este caso, un bloque actual tiene la misma información de movimiento temporal que sus bloques contiguos (por ejemplo, bloques contiguos izquierdo, superior, inferior y/o derecho). Además, el bloque actual y los bloques contiguos se fusionan y ARP se realiza una vez para los bloques fusionados.

[0207] La FIG. 17 es un diagrama de bloques que ilustra un codificador de vídeo 20 de ejemplo que se puede configurar para realizar las técnicas descritas en la presente divulgación. El codificador de vídeo 20 puede realizar la intracodificación y la intercodificación de bloques de vídeo dentro de fragmentos de vídeo. 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 o entrevista para reducir o eliminar la redundancia temporal en el vídeo dentro de tramas o imágenes adyacentes de una secuencia de vídeo. El intramodo (modo I) se puede referir a cualquiera de varios modos de compresión de base espacial. Los intermodos, tales como la predicción unidireccional (modo P) o la bipredicción (modo B), pueden incluir cualquiera de varios modos de compresión de base temporal.

[0208] En el ejemplo de la FIG. 17, el codificador de vídeo 20 incluye una memoria de datos de vídeo 235, una unidad de procesamiento de predicción 241, una memoria intermedia de imágenes decodificadas (DPB) 264, un sumador 251, una unidad de procesamiento de transformada 252, una unidad de procesamiento de cuantificación 255 y una unidad de codificación por entropía 257. La unidad de procesamiento de predicción 241 incluye una unidad de estimación de movimiento 242, una unidad de compensación de movimiento y disparidad 244, una unidad de predicción residual avanzada (ARP) 245 y una unidad de procesamiento de intrapredicción 246. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 también incluye una unidad de procesamiento de cuantificación inversa 259, una unidad de procesamiento de transformada inversa 260 y un sumador 262. También se puede incluir un filtro de eliminación de bloques (no mostrado en la FIG. 17) para filtrar los límites de bloque, para eliminar distorsiones de efecto pixelado del vídeo reconstruido. Si se desea, el filtro de eliminación de bloques filtrará típicamente la salida del sumador 262. También se pueden usar filtros de bucle adicionales (en bucle o tras un bucle), además del filtro de eliminación de bloques.

[0209] En diversos ejemplos, una o más de las unidades de hardware del codificador de vídeo 20 se puede configurar para de realizar las técnicas de esta divulgación. Por ejemplo, la unidad de compensación de movimiento y disparidad 244 y ARP 245 puede realizar las técnicas de esta divulgación, solas o en combinación con otras unidades de codificador de vídeo 20.

[0210] Como se muestra en la FIG. 17, el codificador de vídeo 20 recibe datos de vídeo (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) que se va a codificar. La memoria de datos de vídeo 235 puede almacenar datos de vídeo que se van a codificar mediante los componentes del codificador de vídeo 20. Los datos de vídeo almacenados en la memoria de datos de vídeo 40 se pueden obtener, por ejemplo, a partir de la fuente de vídeo 18. DPB 264 es una memoria intermedia que almacena datos de vídeo de referencia para su uso en la codificación de datos de vídeo por el codificador de vídeo 20 (por ejemplo, en los modos intracodificación o intercodificación, también denominados modos de codificación intrapredicción o interpredicción). La memoria de datos de vídeo 235 y el DPB 264 pueden estar formadas por cualquiera de entre varios dispositivos de memoria, tales como memoria dinámica de acceso aleatorio (DRAM), incluyendo DRAM síncrona (SDRAM), RAM magnetoresistiva (MRAM), RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. La memoria de datos de vídeo 235 y el DPB 264 pueden ser proporcionadas por el mismo dispositivo de memoria o por dispositivos de memoria independientes. En diversos ejemplos, la memoria de datos de vídeo 235 puede estar en un chip con otros componentes del codificador de vídeo 20, o fuera de chip relativo a esos componentes.

[0211] Como se muestra en la FIG. 17, el codificador de vídeo 20 recibe datos de vídeo y divide los datos en bloques de vídeo. Esta división también puede incluir la división en fragmentos, elementos u otras unidades mayores, así como la división de bloques de vídeo, por ejemplo, de acuerdo con una estructura de árbol cuaternario de unas LCU y CU. El codificador de vídeo 20 ilustra, en general, los componentes que codifican los bloques de vídeo dentro de un fragmento de vídeo que se va a codificar. El fragmento se puede dividir en múltiples bloques de vídeo (y, posiblemente, en conjuntos de bloques de vídeo denominados mosaicos).

[0212] La unidad de procesamiento de predicción 241 puede seleccionar uno de una pluralidad de posibles modos de codificación, tal como uno de una pluralidad de modos de intracodificación, o uno de una pluralidad de modos de intercodificación, para el bloque de vídeo actual en base a resultados de error (por ejemplo, la velocidad de codificación y el nivel de distorsión). La unidad de procesamiento de predicción 241 puede proporcionar el bloque intracodificado o intercodificado resultante al sumador 251 para generar datos de bloque residuales y al sumador 262 para reconstruir el bloque codificado para su uso como imagen de referencia.

[0213] La unidad de procesamiento de intrapredicción 246, dentro de la unidad de procesamiento de predicción 241, puede realizar la codificación intrapredictiva del bloque de vídeo actual con respecto a uno o más bloques contiguos en la misma trama o fragmento que el bloque actual que se va a codificar, para proporcionar una compresión espacial. La unidad de estimación de movimiento 242 y la unidad de compensación de movimiento y disparidad 244 en la unidad de procesamiento de predicción 241 realizan la codificación interpredictiva (incluyendo la codificación intervista) del bloque de vídeo actual en relación con uno o más bloques predictivos de una o más imágenes de referencia (incluyendo imágenes de referencia intervista), por ejemplo, para proporcionar compresión temporal y/o intervista.

[0214] La unidad de estimación de movimiento 242 se puede configurar para determinar el modo de interpredicción para un fragmento de vídeo de acuerdo con un patrón predeterminado para una secuencia de vídeo. La unidad de estimación de movimiento 242 y la unidad de compensación de movimiento y disparidad 244 pueden estar altamente integradas, pero se ilustran por separado para propósitos conceptuales. La estimación de movimiento, realizada por la unidad de estimación de movimiento 242, es el proceso de generación de vectores de movimiento, que estiman el movimiento para los bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar la desviación de una PU de un bloque de vídeo dentro de una trama o imagen de vídeo actual relativa a un bloque predictivo dentro de una imagen de referencia.

[0215] Un bloque predictivo es un bloque que se encuentra que coincide estrechamente con la PU del bloque de vídeo que se va a codificar en términos de diferencia de píxeles, lo que se puede determinar por la suma de diferencias absolutas (SAD), la 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 fraccionarios de las imágenes de referencia almacenadas en el DPB 264. Por ejemplo, el codificador de vídeo 20 puede interpolar valores de posiciones de un cuarto de píxel, posiciones de un octavo de píxel u otras posiciones de píxel fraccionario de la imagen de referencia. Por lo tanto, la unidad de estimación de movimiento 242 puede realizar una búsqueda de movimiento relativa a las posiciones de píxel completo y a las posiciones de píxel fraccionario, y emitir un vector de movimiento con una precisión de píxel fraccionario.

[0216] La unidad de estimación de movimiento 242 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 (incluyendo una imagen de referencia temporal o intervista). Como se describió anteriormente, se puede usar un vector de movimiento para la predicción con compensación de movimiento, mientras que se puede usar un vector de movimiento de disparidad para la predicción con compensación de disparidad. La

imagen de referencia se puede seleccionar a partir de una primera lista de imágenes de referencia (Lista 0 o RefPicList0) o una segunda lista de imágenes de referencia (Lista 1 o RefPicList1), cada una de las cuales identifica una o más imágenes de referencia almacenadas en DPB 264. La unidad de estimación de movimiento y disparidad 242 envía el vector de movimiento calculado a la unidad de codificación por entropía 257 y a la unidad de compensación de movimiento y disparidad 244.

[0217] La compensación de movimiento y/o disparidad, realizada por la unidad de compensación de movimiento y disparidad 244, puede implicar extraer o generar el bloque predictivo en base al vector de movimiento determinado mediante estimación de movimiento, realizando posiblemente interpolaciones hasta una precisión de sub-píxel. Tras recibir el vector de movimiento para la PU del bloque de vídeo actual, la unidad de compensación de movimiento y disparidad 244 puede localizar el bloque predictivo al que apunta el vector de movimiento en una de las listas de imágenes de referencia. El codificador de vídeo 20 forma un bloque de vídeo residual restando los valores de píxel del bloque predictivo a los valores de píxel del bloque de vídeo actual que se esté codificando, formando valores de diferencia de píxel. Los valores de diferencia de píxel forman datos residuales para el bloque, y pueden incluir componentes de diferencia tanto de luma como de croma. El sumador 251 representa el componente o los componentes que realizan esta operación de resta. La unidad de compensación de movimiento y disparidad 244 también puede generar elementos sintácticos asociados a los bloques de vídeo y al fragmento de vídeo para su uso por el descodificador de vídeo 30 en la descodificación de los bloques de vídeo del fragmento de vídeo.

[0218] El codificador de vídeo 20, que incluye la unidad ARP 245 y la unidad de compensación de movimiento y disparidad 244, puede realizar cualquiera de las técnicas de predicción bidireccional y ARP, por ejemplo, técnicas de entrevista o ARP temporal descritas en el presente documento. En particular, en un ejemplo de la divulgación, el codificador de vídeo se puede configurar para codificar un bloque actual de datos de vídeo usando predicción bidireccional y ARP entrevista. Para el bloque actual de datos de vídeo, la unidad de compensación de movimiento y disparidad 244 se puede configurar para determinar información de movimiento temporal para una primera dirección de predicción (por ejemplo, lista de imágenes de referencia X) para el bloque actual de datos de vídeo e identificar bloques de referencia para una segunda dirección de predicción (por ejemplo, lista de imágenes de referencia Y) usando la información de movimiento temporal determinada para la primera dirección de predicción, en la que los bloques de referencia para la segunda dirección de predicción están en una unidad de acceso diferente al bloque actual de datos de vídeo. De esta manera, se necesitan menos accesos a memoria de información de movimiento y bloques de referencia para codificar el bloque de vídeo actual.

[0219] La unidad de procesamiento de intrapredicción 246 puede realizar la intrapredicción en un bloque actual, como alternativa a la interpredicción realizada por la unidad de estimación de movimiento 242 y la unidad de compensación de movimiento y disparidad 244, como se ha descrito anteriormente. En particular, la unidad de procesamiento de intrapredicción 246 puede determinar un modo de intrapredicción para su uso en la codificación de un bloque actual. En algunos ejemplos, la unidad de procesamiento de intrapredicción 246 puede codificar un bloque de vídeo actual usando diversos modos de intrapredicción, por ejemplo, durante pases de codificación separadas, y el módulo de procesamiento de intrapredicción 246 (o la unidad de procesamiento de predicción 241, en algunos ejemplos) puede seleccionar un modo adecuado de intrapredicción a usar, entre los modos sometidos a prueba.

[0220] Por ejemplo, la unidad de procesamiento de intrapredicción 246 puede calcular valores de velocidad-distorsión usando un análisis de velocidad-distorsión para los diversos modos de intrapredicción probados, y seleccionar el modo de intrapredicción que tenga las mejores características de velocidad-distorsión entre los modos probados. El análisis de velocidad-distorsión determina, en general, una cantidad de distorsión (o error) entre un bloque codificado y un bloque original, no codificado, que se codificó para producir el bloque codificado, así como una velocidad de bits (es decir, un número de bits) usada para producir el bloque codificado. La unidad de procesamiento de intrapredicción 246 puede calcular relaciones a partir de las distorsiones y velocidades para los diversos bloques codificados, para determinar qué modo de intrapredicción presenta el mejor valor de velocidad-distorsión para el bloque.

[0221] En cualquier caso, tras seleccionar un modo intrapredicción para un bloque, la unidad de procesamiento de intrapredicción 246 puede proporcionar información que indica el modo intrapredicción seleccionado para el bloque, a la unidad de codificación por entropía 257. La unidad de codificación por entropía 257 puede codificar la información que indique el modo de intrapredicción seleccionado de acuerdo con las técnicas de la presente divulgación. El codificador de vídeo 20 puede incluir, en el flujo de bits transmitido, datos de configuración, que pueden incluir una pluralidad de tablas de índices de modo de intrapredicción y una pluralidad de tablas de índices de modo de intrapredicción modificadas (también denominadas tablas de correlación de palabras de código), definiciones de contextos de codificación para diversos bloques e indicaciones de un modo de intrapredicción más probable, una tabla de índices de modo de intrapredicción y una tabla de índices de modo de intrapredicción modificada para su uso para cada uno de los contextos.

[0222] Después de que la unidad de procesamiento de predicción 241 genera el bloque predictivo para el bloque de vídeo actual, por medio de interpredicción o bien intrapredicción, el codificador de vídeo 20 forma un bloque de vídeo residual restando el bloque predictivo al bloque de vídeo actual. Los datos de vídeo residuales del bloque residual se pueden incluir en una o más TU y aplicarse a la unidad de procesamiento de transformada 252. La unidad de procesamiento de transformada 252 transforma los datos de vídeo residuales en coeficientes de transformada

residuales usando una transformada, tal como una transformada de coseno discreta (DCT) o una transformada conceptualmente similar. La unidad de procesamiento de transformada 252 puede convertir los datos de vídeo residuales de un dominio de píxel a un dominio de transformada, tal como un dominio de frecuencia.

[0223] La unidad de procesamiento de transformada 252 puede enviar los coeficientes de transformada resultantes a la unidad de procesamiento de cuantificación 255. La unidad de procesamiento de cuantificación 255 cuantifica los coeficientes de transformada para reducir aún más la velocidad de transmisión de bits. El proceso de cuantificación puede reducir la profundidad de bits asociada con algunos, o con la totalidad, de los coeficientes. El grado de cuantificación se puede modificar ajustando un parámetro de cuantificación. En algunos ejemplos, la unidad de procesamiento de cuantificación 255 luego puede realizar una exploración de la matriz, incluyendo los coeficientes de transformada cuantificados. De forma alternativa, la unidad de codificación por entropía 257 puede realizar la exploración.

[0224] Tras la cuantificación, la unidad de codificación por entropía 257 codifica por entropía los coeficientes de transformada cuantificados. Por ejemplo, la unidad de codificación por entropía 257 puede realizar una codificación de longitud variable adaptativa al contexto (CAVLC), una codificación aritmética binaria adaptativa al contexto (CABAC), una codificación aritmética binaria adaptativa al contexto basada en la sintaxis (SBAC), una codificación por entropía por división de intervalos de probabilidad (PIPE) u otra metodología o técnica de codificación por entropía. Tras la codificación por entropía realizada por la unidad de codificación por entropía 257, el flujo de bits de vídeo codificado se puede transmitir al descodificador de vídeo 30, o archivar para su posterior transmisión o recuperación mediante el descodificador de vídeo 30. La unidad de codificación por entropía 257 también puede codificar por entropía los vectores de movimiento y los otros elementos sintácticos para el fragmento de vídeo actual que se esté codificando.

[0225] La unidad de cuantificación inversa 259 y la unidad de procesamiento de transformada inversa 260 aplican una cuantificación inversa y una transformación inversa, respectivamente, para reconstruir el bloque residual en el dominio de los píxeles, para su posterior uso como bloque de referencia de una imagen de referencia. La unidad de compensación de movimiento y disparidad 244 puede calcular un bloque de referencia sumando el bloque residual a un bloque predictivo de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. La unidad de compensación de movimiento y disparidad 244 también puede aplicar uno o más filtros de interpolación al bloque residual reconstruido para calcular valores de píxeles fraccionarios, para su uso en la estimación de movimiento. El sumador 262 añade el bloque residual reconstruido al bloque de predicción compensado por movimiento producido por la unidad de compensación de movimiento y disparidad 244 para producir un bloque de referencia para su almacenamiento en DPB 264. El bloque de referencia puede ser usado por la unidad de estimación de movimiento 242 y la unidad de compensación de movimiento y disparidad 244 como un bloque de referencia para realizar una interpredicción de un bloque en una trama o imagen de vídeo posterior.

[0226] La FIG. 18 es un diagrama de bloques que ilustra un descodificador de vídeo 30 de ejemplo que puede implementar las técnicas descritas en la presente divulgación. En el ejemplo de la FIG. 18, el descodificador de vídeo 30 incluye una memoria de datos de vídeo 278, una unidad de descodificación por entropía 280, una unidad de procesamiento de predicción 281, una unidad de cuantificación inversa 286, una unidad de procesamiento de transformada inversa 288, un sumador 291 y una memoria intermedia de imágenes descodificadas (DPB) 292. La unidad de procesamiento de predicción 281 incluye la unidad de compensación de movimiento y disparidad 282, la unidad ARP 283 y la unidad de procesamiento de intrapredicción 284. En algunos ejemplos, el descodificador de vídeo 30 puede realizar una pasada de descodificación generalmente recíproca a la pasada de codificación descrita con respecto al codificador de vídeo 20 de la FIG. 17.

[0227] En diversos ejemplos, una o más unidades de hardware de descodificador de vídeo 30 se les puede encargar la realización de las técnicas de esta divulgación. Por ejemplo, la unidad ARP 283 y la unidad de compensación de movimiento y disparidad 282 pueden realizar las técnicas de esta divulgación, solas o en combinación con otras unidades del codificador de vídeo.

[0228] La memoria de datos de vídeo 278 puede almacenar datos de vídeo, tales como un flujo de bits de vídeo codificado, que se va a descodificar por los componentes del descodificador de vídeo 30. Los datos de vídeo almacenados en la memoria de datos de vídeo 278 se pueden obtener desde una fuente de vídeo local, tal como una cámara, por medio de la comunicación de datos de vídeo de red cableada o inalámbrica, o accediendo a medios físicos de almacenamiento de datos. La memoria de datos de vídeo 278 puede formar un búfer de imágenes codificadas (CPB) que almacena datos de vídeo codificados a partir de un flujo de bits de vídeo codificado. DPB 292 es un ejemplo de DPB que almacena datos de vídeo de referencia para su uso en la descodificación de datos de vídeo por el descodificador de vídeo 30 (por ejemplo, en los modos intracodificación o intercodificación, también denominados modos de codificación intrapredicción o interpredicción). La memoria de datos de vídeo 278 y el DPB 292 pueden estar formadas por cualquiera de entre varios dispositivos de memoria, tales como memoria dinámica de acceso aleatorio (DRAM), incluyendo DRAM síncrona (SDRAM), RAM magnetoresistiva (MRAM), RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. La memoria de datos de vídeo 278 y el DPB 292 pueden ser proporcionadas por el mismo dispositivo de memoria o por dispositivos de memoria independientes. En diversos ejemplos, la memoria

de datos de vídeo 278 puede estar en un chip con otros componentes del descodificador de vídeo 30, o fuera de chip relativo a esos componentes.

[0229] Durante el procedimiento de descodificación, el descodificador de vídeo 30 recibe un flujo de bits de vídeo codificado que representa los bloques de vídeo de un fragmento de vídeo codificado y elementos sintácticos asociados desde el codificador de vídeo 20. La unidad de descodificación por entropía 280 del descodificador de vídeo 30 descodifica por entropía el flujo de bits para generar coeficientes cuantificados, vectores de movimiento y otros elementos sintácticos. La unidad de descodificación por entropía 280 reenvía los vectores de movimiento y otros elementos sintácticos a la unidad de procesamiento de predicción 281. El descodificador de vídeo 30 puede recibir los elementos de sintaxis a nivel de fragmento de vídeo y/o a nivel de bloque de vídeo.

[0230] Cuando el fragmento de vídeo se codifica como un fragmento intracodificado (I), la unidad de procesamiento de intrapredicción 284 de la unidad de procesamiento de predicción 281 puede generar datos de predicción para un bloque de vídeo del fragmento de vídeo actual en base a un modo de intrapredicción señalizado y datos de los bloques previamente descodificados de la trama o imagen actual. Cuando la trama de vídeo se codifica como un fragmento intercodificado (es decir, B o P), o fragmento codificado entrevista, la unidad de compensación de movimiento y disparidad 282 de la unidad de procesamiento de predicción 281 produce bloques predictivos para un bloque de vídeo del fragmento de vídeo actual, en base a los vectores de movimiento, vectores de disparidad y otros elementos sintácticos recibidos desde la unidad de descodificación por entropía 280. Los bloques predictivos se pueden producir a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia incluyendo imágenes de referencia entrevista. El descodificador de vídeo 30 puede construir las listas de tramas de referencia, RefPicList0 y RefPicList1 usando técnicas de construcción predeterminada o cualquier otra técnica en base a las imágenes de referencia almacenadas en DPB 292.

[0231] La unidad de compensación de movimiento y disparidad 282 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 producir los bloques predictivos para el bloque de vídeo actual que está siendo descodificado. Por ejemplo, la unidad de compensación de movimiento y disparidad 282 usa algunos de los elementos sintácticos recibidos para determinar un modo de predicción (por ejemplo, intra o interpredicción) usado para codificar los bloques de vídeo del fragmento de vídeo, un tipo de fragmento de interpredicción (por ejemplo, un fragmento B, un fragmento P y/o un fragmento predicho entrevista), información de construcción para una o más de las listas de imágenes de referencia para el fragmento, vectores de movimiento para cada bloque de vídeo intercodificado del fragmento, el estado de interpredicción para cada bloque de vídeo intercodificado del fragmento y otra información para descodificar los bloques de vídeo en el fragmento de vídeo actual.

[0232] La unidad de compensación de movimiento y disparidad 282 también puede realizar la interpolación en base a filtros de interpolación. La unidad de compensación de movimiento y disparidad 282 puede usar filtros de interpolación como los usados por el codificador de vídeo 20 durante la codificación de los bloques de vídeo, para calcular valores interpolados para píxeles fraccionarios de bloques de referencia. En este caso, la unidad de compensación de movimiento y disparidad 282 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.

[0233] La unidad de cuantificación inversa 286 cuantifica de manera inversa (es decir, descuantifica), los coeficientes de transformada cuantificados, proporcionados en el flujo de bits y descodificados por la unidad de descodificación por entropía 280. El proceso de cuantificación inversa puede incluir el uso de un parámetro de cuantificación calculado por el codificador de vídeo 20 para cada bloque de vídeo del fragmento de vídeo para determinar un grado de cuantificación y, del mismo modo, un grado de cuantificación inversa que se debería aplicar. La unidad de procesamiento de transformada inversa 288 aplica una transformada inversa, por ejemplo, una DCT inversa, una transformada de entero inversa o un procedimiento de transformada inversa conceptualmente similar, a los coeficientes de transformada para producir bloques residuales en el dominio de píxel.

[0234] El descodificador de vídeo 30, que incluye la unidad ARP 283 y la unidad de compensación de movimiento y disparidad 282, puede realizar cualquiera de las técnicas de predicción bidireccional y/o ARP, por ejemplo, técnicas de ARP entrevista o temporal descritas en este documento. En particular, en un ejemplo de la divulgación, el descodificador de vídeo 30 puede recibir un bloque actual de datos de vídeo codificados usando predicción bidireccional y ARP entrevista. El bloque codificado de datos de vídeo se puede almacenar en la memoria de datos 278. Para el bloque codificado de datos de vídeo, la unidad de compensación de movimiento y disparidad 282 se puede configurar para determinar información de movimiento temporal para una primera dirección de predicción (por ejemplo, lista de imágenes de referencia para el bloque codificado de datos de vídeo e identificar bloques de referencia para una segunda dirección de predicción (por ejemplo, lista de imágenes de referencia Y) usando la información de movimiento temporal determinada para la primera dirección de predicción, en la que los bloques de referencia para la segunda dirección de predicción están en una unidad de acceso diferente al bloque actual de datos de vídeo. De esta manera, se necesitan menos accesos a memoria de información de movimiento y bloques de referencia para descodificar el bloque de vídeo codificado.

[0235] Después de que la unidad de compensación de movimiento y disparidad 282 genera el bloque predictivo para el bloque de vídeo actual, en base a los vectores de movimiento y otros elementos sintácticos, el descodificador de vídeo 30 forma un bloque de vídeo descodificado sumando los bloques residuales procedentes de la unidad de procesamiento de transformada inversa 288 a los correspondientes bloques predictivos generados por la unidad de compensación de movimiento y disparidad 282. El sumador 291 representa el componente o los componentes que realizan esta operación de suma. Si se desea, también se puede aplicar un filtro de eliminación de bloques para filtrar los bloques descodificados a fin de eliminar distorsiones de efecto pixelado. También se pueden usar otros filtros de bucle (bien en el bucle de codificación o bien después del bucle de codificación) para suavizar las transiciones entre píxeles o mejorar de otro modo la calidad de vídeo. Los bloques de vídeo descodificados en una trama o imagen dada son a continuación almacenados en la DPB 292, que almacena imágenes de referencia usadas para la posterior compensación de movimiento. La DPB 292 almacena también vídeo descodificado para su presentación posterior en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la FIG. 1.

[0236] La FIG. 19 es un diagrama de flujo que ilustra un ejemplo de procedimiento de ARP para codificar un bloque de vídeo de acuerdo con las técnicas descritas en esta divulgación. Las técnicas de la FIG. 19 se pueden realizar mediante cualquier combinación de estructuras de hardware del codificador de vídeo 20, incluyendo la unidad de compensación de movimiento y disparidad 244 y la unidad ARP 245.

[0237] En un ejemplo de la divulgación, el codificador de vídeo 20 se puede configurar para codificar un bloque de datos de vídeo usando ARP y predicción bidireccional. En este ejemplo, la predicción bidireccional incluye predicción temporal para una primera dirección de predicción (por ejemplo, para la lista de imágenes de referencia X) y predicción entrevista para una segunda dirección de predicción (por ejemplo, para la lista de imágenes de referencia Y). La unidad de compensación de movimiento y disparidad 244 se puede configurar para determinar la información de movimiento temporal para la primera dirección de predicción para el bloque de datos de vídeo (1900). La unidad ARP 245 se puede configurar para identificar bloques de referencia para la primera dirección de predicción usando la información de movimiento temporal determinada (1910) e identificar bloques de referencia para la segunda dirección de predicción, diferente de la primera dirección de predicción, usando la información de movimiento temporal determinada para la primera dirección de predicción (1920). Los bloques de referencia pueden estar en una unidad de acceso diferente a la unidad de acceso para el bloque de datos de vídeo. La unidad ARP 245 se puede configurar además para realizar predicción residual avanzada en el bloque de datos de vídeo usando los bloques de referencia identificados para la primera dirección de predicción y la segunda dirección de predicción (1930).

[0238] En otros ejemplos de la divulgación, la unidad de compensación de movimiento y disparidad 244 se puede configurar para determinar la información de movimiento de disparidad para la segunda dirección de predicción para el primer bloque codificado de datos de vídeo. Además, la unidad ARP 245 se puede configurar para identificar un primer bloque de referencia para la primera dirección de predicción usando la información de movimiento temporal determinada, en el que el primer bloque de referencia está en una segunda unidad de acceso de la primera vista. La unidad ARP 245 se puede configurar además para identificar un segundo bloque de referencia para la segunda dirección de predicción usando la información de movimiento temporal determinada, y para identificar un tercer bloque de referencia para la segunda dirección de predicción usando la información de movimiento temporal determinada y la información de movimiento de disparidad determinada, en el que el tercer bloque de referencia está en una tercera unidad de acceso de una segunda vista.

[0239] La FIG.20 es un diagrama de flujo que ilustra un ejemplo de procedimiento ARP para descodificar un bloque de vídeo de acuerdo con las técnicas descritas en esta divulgación. Las técnicas de la FIG. 20 se puede realizar mediante cualquier combinación de estructuras de hardware de descodificador de vídeo, unidad ARP 283 y unidad de compensación de movimiento y disparidad 282.

[0240] En un ejemplo de la divulgación, el descodificador de vídeo 30 se puede configurar para almacenar un primer bloque codificado de datos de vídeo en una primera unidad de acceso de una primera vista, en el que el primer bloque codificado de datos de vídeo fue codificado usando predicción residual avanzada y predicción bidireccional (2000). La predicción bidireccional puede incluir una predicción temporal para una primera dirección de predicción y una predicción entrevista para una segunda dirección de predicción.

[0241] La unidad de compensación de movimiento y disparidad 282 se puede configurar para determinar información de movimiento temporal para la primera dirección de predicción para el primer bloque codificado de datos de vídeo (2010). La unidad ARP 283 se puede configurar para determinar la información de movimiento de disparidad para la segunda dirección de predicción para el primer bloque codificado de datos de vídeo (2020), e identificar bloques de referencia para la segunda dirección de predicción, diferente de la primera dirección de predicción, usando la información de movimiento temporal determinado para la primera dirección de predicción (2030). Los bloques de referencia pueden estar en una unidad de acceso diferente a la primera unidad de acceso. La unidad ARP 283 se puede configurar además para realizar una predicción residual avanzada en el primer bloque codificado de datos de vídeo usando los bloques de referencia identificados para la segunda dirección de predicción (2040).

[0242] En otro ejemplo de la divulgación, la unidad ARP 238 se puede configurar para identificar bloques de referencia para la primera dirección de predicción usando la información de movimiento temporal determinada, y realizando una

predicción residual avanzada en el primer bloque codificado de datos de vídeo usando los bloques de referencia identificados para la primera dirección de predicción. La unidad ARP 283 se puede configurar además para identificar un segundo bloque de referencia para la segunda dirección de predicción usando la información de movimiento temporal determinada, y para identificar un tercer bloque de referencia para la segunda dirección de predicción usando la información de movimiento temporal determinada y la información de movimiento de disparidad determinada, en el que el tercer bloque de referencia está en una tercera unidad de acceso de una segunda vista. El primer bloque de referencia de la primera dirección de predicción es el mismo que el segundo bloque de referencia de la segunda dirección de predicción

[0243] En otro ejemplo de la divulgación, el decodificador de vídeo 30 se puede configurar para decodificar el primer bloque codificado de datos de vídeo usando los bloques de referencia identificados para la primera dirección de predicción y los bloques de referencia identificados para la segunda dirección de predicción.

[0244] En otro ejemplo de la divulgación, el decodificador de vídeo 30 se puede configurar para decodificar el primer bloque codificado de datos de vídeo usando una de predicción residual avanzada a nivel de bloque o predicción residual avanzada a nivel de unidad de predicción para producir datos de vídeo residuales, decodificar los datos residuales usando predicción bidireccional, el bloque de referencia identificado para la primera dirección de predicción y el bloque de referencia identificado para la segunda dirección de predicción para producir un bloque decodificado de datos de vídeo.

[0245] En otro ejemplo de la divulgación, el decodificador de vídeo 30 se puede configurar además para almacenar un segundo bloque codificado de datos de vídeo en una cuarta unidad de acceso de una tercera vista, en el que el segundo bloque codificado de datos de vídeo se codificó usando predicción residual avanzada y predicción bidireccional. La predicción bidireccional puede incluir una predicción temporal para una tercera dirección de predicción y una predicción entrevista para una cuarta dirección de predicción.

[0246] La unidad de compensación de movimiento y disparidad 282 se puede configurar para determinar información de movimiento temporal para la primera dirección de predicción para el primer bloque codificado de datos de vídeo. La unidad ARP 283 se puede configurar para identificar bloques de referencia para la primera dirección de predicción usando la información de movimiento temporal determinada. La unidad ARP 283 puede identificar además bloques de referencia para la segunda dirección de predicción, diferente de la primera dirección de predicción, usando la información de movimiento temporal determinada para la primera dirección de predicción, en la que los bloques de referencia están en una unidad de acceso diferente a la primera unidad de acceso. La unidad ARP 283 también puede realizar una predicción residual avanzada en el primer bloque codificado de datos de vídeo usando los bloques de referencia identificados para la primera dirección de predicción y la segunda dirección de predicción.

[0247] En otro ejemplo de la divulgación, la unidad de compensación de movimiento y disparidad 282 se puede configurar para determinar la información de movimiento de disparidad para la segunda dirección de predicción para el primer bloque codificado de datos de vídeo. La unidad ARP 283 se puede configurar para identificar un primer bloque de referencia para la primera dirección de predicción usando la información de movimiento temporal determinada, en el que el primer bloque de referencia está en una segunda unidad de acceso de la primera vista. La unidad ARP 283 se puede configurar además para identificar un segundo bloque de referencia para la segunda dirección de predicción usando la información de movimiento temporal determinada y la información de movimiento de disparidad determinada, en el que el segundo bloque de referencia está en una tercera unidad de acceso de una segunda vista.

[0248] En otro ejemplo de la divulgación, el decodificador de vídeo 30 se puede configurar para decodificar el primer bloque codificado de datos de vídeo usando los bloques de referencia identificados para la primera dirección de predicción y los bloques de referencia identificados para la segunda dirección de predicción. El decodificador de vídeo 30 se puede configurar además para decodificar el primer bloque codificado de datos de vídeo usando una predicción residual avanzada a nivel de bloque o predicción residual avanzada a nivel de unidad de predicción para producir datos de vídeo residuales, y decodificar los datos residuales usando predicción bidireccional, el bloque de referencia identificado para la primera dirección de predicción y el bloque de referencia identificado para la segunda dirección de predicción para producir un bloque decodificado de datos de vídeo.

[0249] En algunos ejemplos, uno o más aspectos de las técnicas descritas en esta divulgación pueden ser realizados por un dispositivo de red intermedio, tal como un elemento de red sensible a los medios (MANE), un procesador de adaptación de flujo, un procesador de empalme o un procesador de edición. Por ejemplo, dicho dispositivo intermedio se puede configurar para generar o recibir cualquiera de una variedad de señales como se describe en esta divulgación.

[0250] En uno o más ejemplos, las funciones descritas se pueden implementar en hardware, programa informático, firmware o cualquier combinación de los mismos. Si se implementan en software, las funciones se pueden almacenar en o transmitir a través de un medio legible por ordenador como una o más instrucciones o código, y ejecutarse por 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, tales como medios de almacenamiento

de datos, o medios de comunicación que incluyan cualquier medio que facilite la transferencia de un programa informático de un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador pueden corresponder, en general, a (1) medios de almacenamiento tangibles legibles por ordenador que sean no transitorios o a (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que se pueda acceder 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.

[0251] A modo de ejemplo, y no de limitación, dichos medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento de disco óptico, almacenamiento de disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que se pueda usar para almacenar un código de programa deseado en forma de instrucciones o estructuras de datos y al que se pueda acceder mediante un ordenador. Además, cualquier conexión recibe apropiadamente la denominación de medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, un servidor u otra fuente remota usando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas están incluidos en la definición de medio. Sin embargo, se debe entender que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales ni otros medios transitorios, sino que, en cambio, se refieren a medios de almacenamiento tangibles no transitorios. El término disco, como se usa en el presente documento, incluye disco compacto (CD), disco láser, disco óptico, disco versátil digital (DVD), disco flexible y disco Blu-ray, donde unos discos reproducen normalmente los datos magnéticamente, mientras que otros discos reproducen datos ópticamente con láseres. Las combinaciones de lo anterior también se deben incluir dentro del alcance de los medios legibles por ordenador.

[0252] Las instrucciones se pueden ejecutar por uno o más procesadores, tales como uno o más procesadores digitales de señales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), matrices lógicas programables in situ (FPGA) u otros circuitos lógicos discretos o integrados 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 de programa informático dedicados configurados para la codificación y la decodificación, o incorporarse en un códec combinado. Además, las técnicas se podrían implementar por completo en uno o más circuitos o elementos lógicos.

[0253] Las técnicas de esta divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluyendo un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En esta divulgación se describen diversos componentes, módulos o unidades para destacar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no se requiere necesariamente su realización mediante diferentes unidades de hardware. En su lugar, como se describe anteriormente, diversas unidades se pueden combinar en una unidad de hardware de códec o proporcionar mediante un grupo de unidades de hardware interoperativas, que incluya uno o más procesadores como se describe anteriormente, junto con software y/o firmware adecuados. Se han descrito diversos ejemplos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.

REIVINDICACIONES

1. Un procedimiento para decodificar datos de vídeo multivista, comprendiendo el procedimiento:

- 5 recibir un primer bloque codificado de datos de vídeo (250) en una primera unidad de acceso de una primera vista (Vm), en el que el primer bloque codificado de datos de vídeo (250) fue codificado usando predicción residual avanzada y predicción bidireccional, incluyendo la predicción bidireccional predicción temporal para una primera dirección de predicción y predicción entrevista para una segunda dirección de predicción, y en el que la predicción residual avanzada incluye realizar una predicción de un residual del primer bloque codificado de datos de vídeo en relación con un residual de un bloque predictor residual;
- 10 determinar información de movimiento temporal (TMV, 210) para la primera dirección de predicción para el primer bloque codificado de datos de vídeo (250);
- 15 identificar un primer bloque de referencia temporal (212) para la primera dirección de predicción usando la primera información de movimiento temporal (TMV, 210) para la primera dirección de predicción, en el que el primer bloque de referencia temporal (212) está en la primera vista (Vm) y en una segunda unidad de acceso que es diferente a la primera unidad de acceso;
- 20 derivar un vector de disparidad (DV, 204) para la segunda dirección de predicción para el primer bloque codificado de datos de vídeo (250);
- 25 identificar un primer bloque de referencia entrevista (206) usando el vector de disparidad (DV, 204) para la segunda dirección de predicción, en el que el primer bloque de referencia entrevista (206) está en la primera unidad de acceso y está en una segunda vista (V0) que es diferente a la primera vista (Vm);
- 30 identificar un segundo bloque de referencia entrevista (216) usando el primer bloque de referencia entrevista (206) y la primera información de movimiento temporal (TMV, 210) para la primera dirección de predicción, en el que el segundo bloque de referencia entrevista (216) está en la segunda vista (V0) y en la segunda unidad de acceso;
- 35 determinar un primer bloque predictor residual a partir de la diferencia píxel a píxel entre el primer bloque de referencia entrevista (206) y el segundo bloque de referencia entrevista (216);
- 40 determinar información de movimiento de disparidad (DMV, 254) para la segunda dirección de predicción para el primer bloque codificado de datos de vídeo (250);
- 45 identificar un tercer bloque de referencia entrevista (256) usando la primera información de movimiento de disparidad (DMV, 254) para la segunda dirección de predicción, en el que el tercer bloque de referencia entrevista (256) está en la segunda vista (V0) y en la primera unidad de acceso;
- 50 identificar un cuarto bloque de referencia entrevista (290) usando el tercer bloque de referencia entrevista (256) y la primera información de movimiento temporal (TMV, 210) para la primera dirección de predicción, en el que el cuarto bloque de referencia entrevista (290) está en la segunda vista (V0) y en la segunda unidad de acceso;
- 55 determinar un segundo bloque predictor residual a partir de la diferencia píxel a píxel entre el primer bloque de referencia temporal (212) y el cuarto bloque de referencia entrevista (290); y
- 60 descodificar el primer bloque codificado de datos de vídeo (250) realizando la predicción residual avanzada en el primer bloque codificado de datos de vídeo (250) usando el primer bloque de referencia temporal (212), el primer bloque predictor residual y el segundo bloque predictor residual.
2. El procedimiento de la reivindicación 1, en el que los datos de vídeo son datos de vídeo de codificación de vídeo de alta eficacia multivista, MC-HEVC.
3. El procedimiento de la reivindicación 1, en el que los datos de vídeo son datos de vídeo multivista más profundidad.
4. El procedimiento de la reivindicación 3, en el que los datos de vídeo son datos de vídeo de codificación de vídeo de alta eficacia 3D, 3D-HEVC.
5. El procedimiento de la reivindicación 1, en el que la segunda vista (V0) es una vista de base.
6. Un aparato configurado para decodificar datos de vídeo multivista, comprendiendo el aparato:

una memoria de datos de vídeo configurada para almacenar un primer bloque codificado de datos de vídeo (250) en una primera unidad de acceso de una primera vista (Vm), en el que el primer bloque codificado de datos de vídeo (250) fue codificado usando predicción residual avanzada y predicción bidireccional, incluyendo la predicción bidireccional predicción temporal para una primera dirección de predicción y predicción entrevista para una segunda dirección de predicción y en el que la predicción residual avanzada incluye realizar una predicción de un residual del primer bloque codificado de datos de vídeo en relación con un residual de un bloque predictor residual; y

uno o más procesadores en comunicación con la memoria de datos de vídeo y configurados para:

determinar información de movimiento temporal (TMV, 210) para la primera dirección de predicción para el primer bloque codificado de datos de vídeo (250);

identificar un primer bloque de referencia temporal (212) para la primera dirección de predicción usando la primera información de movimiento temporal (TMV, 210) para la primera dirección de predicción, en el que el primer bloque de referencia temporal (212) está en la primera vista (Vm) y en una segunda unidad de acceso que es diferente a la primera unidad de acceso;

derivar un vector de disparidad (DV, 204) para la segunda dirección de predicción para el primer bloque codificado de datos de vídeo (250);

identificar un primer bloque de referencia entrevista (206) usando el vector de disparidad (DV, 204) para la segunda dirección de predicción, en el que el primer bloque de referencia entrevista (206) está en la primera unidad de acceso y está en una segunda vista (V0) que es diferente a la primera vista (Vm);

identificar un segundo bloque de referencia entrevista (216) usando el primer bloque de referencia entrevista (206) y la primera información de movimiento temporal (TMV, 210) para la primera dirección de predicción, en el que el segundo bloque de referencia entrevista (216) está en la segunda vista (V0) y en la segunda unidad de acceso;

determinar un primer bloque predictor residual a partir de la diferencia píxel a píxel entre el primer bloque de referencia entrevista (206) y el segundo bloque de referencia entrevista (216);

determinar información de movimiento de disparidad (DMV, 254) para la segunda dirección de predicción para el primer bloque codificado de datos de vídeo (250);

identificar un tercer bloque de referencia entrevista (256) usando la primera información de movimiento de disparidad (DMV, 254) para la segunda dirección de predicción, en el que el tercer bloque de referencia entrevista (256) está en la segunda vista (V0) y en la primera unidad de acceso;

identificar un cuarto bloque de referencia entrevista (290) usando el tercer bloque de referencia entrevista (256) y la primera información de movimiento temporal (TMV, 210) para la primera dirección de predicción, en el que el cuarto bloque de referencia entrevista (290) está en la segunda vista (V0) y en la segunda unidad de acceso;

determinar un segundo bloque predictor residual a partir de la diferencia píxel a píxel entre el primer bloque de referencia temporal (212) y el cuarto bloque de referencia entrevista (290); y

descodificar el primer bloque codificado de datos de vídeo (250) realizando una predicción residual avanzada en el primer bloque codificado de datos de vídeo (250) usando el primer bloque predictor residual y el segundo bloque predictor residual.

7. El aparato de la reivindicación 6, que comprende además una pantalla configurada para mostrar el bloque descodificado de datos de vídeo.

8. El aparato de la reivindicación 6, en el que la memoria de datos de vídeo y el uno o más procesadores están alojados dentro de un ordenador de escritorio, un ordenador tipo notebook, un ordenador portátil, un descodificador, un auricular de teléfono, un teléfono inteligente, una almohadilla inteligente, una tableta, un televisor, una cámara, un reproductor de medios digitales, una consola de videojuegos o un dispositivo de transmisión continua de vídeo.

9. El aparato de la reivindicación 6, en el que los datos de vídeo son datos de vídeo de codificación de vídeo de alta eficacia multivista, MC-HEVC.

10. El aparato de la reivindicación 6, en el que los datos de vídeo son datos de vídeo multivista más profundidad.

11. El aparato de la reivindicación 10, en el que los datos de vídeo son datos de vídeo de codificación de vídeo de alta eficacia 3D, 3D-HEVC.
12. El aparato de la reivindicación 6, en el que la segunda vista (V0) es una vista de base.
13. Un medio de almacenamiento legible por ordenador que almacena instrucciones que, cuando se ejecutan, hacen que uno o más procesadores de un dispositivo configurado para descodificar datos de vídeo lleven a cabo un procedimiento de acuerdo con cualquiera de las reivindicaciones 1 a 5.

5

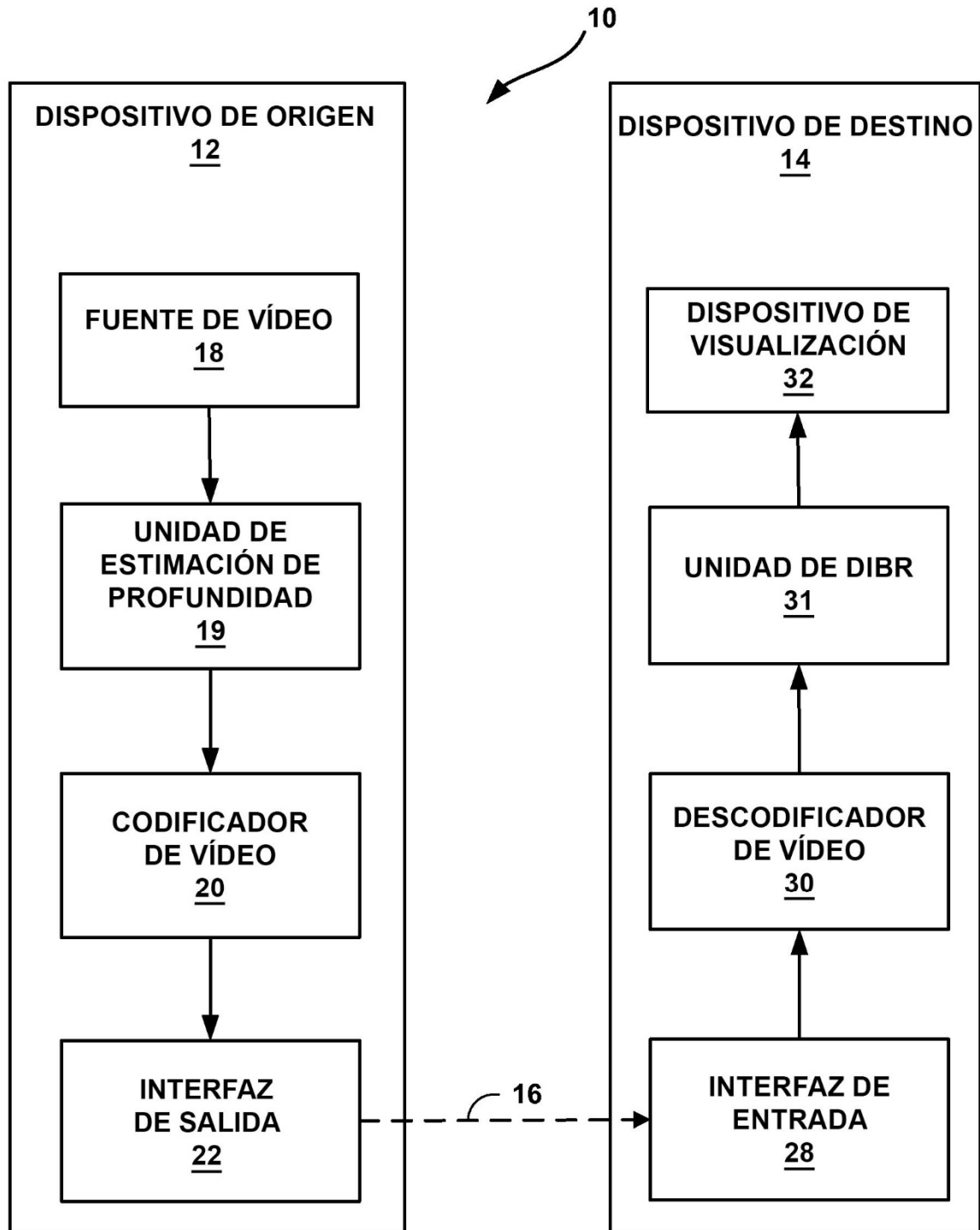


FIG. 1

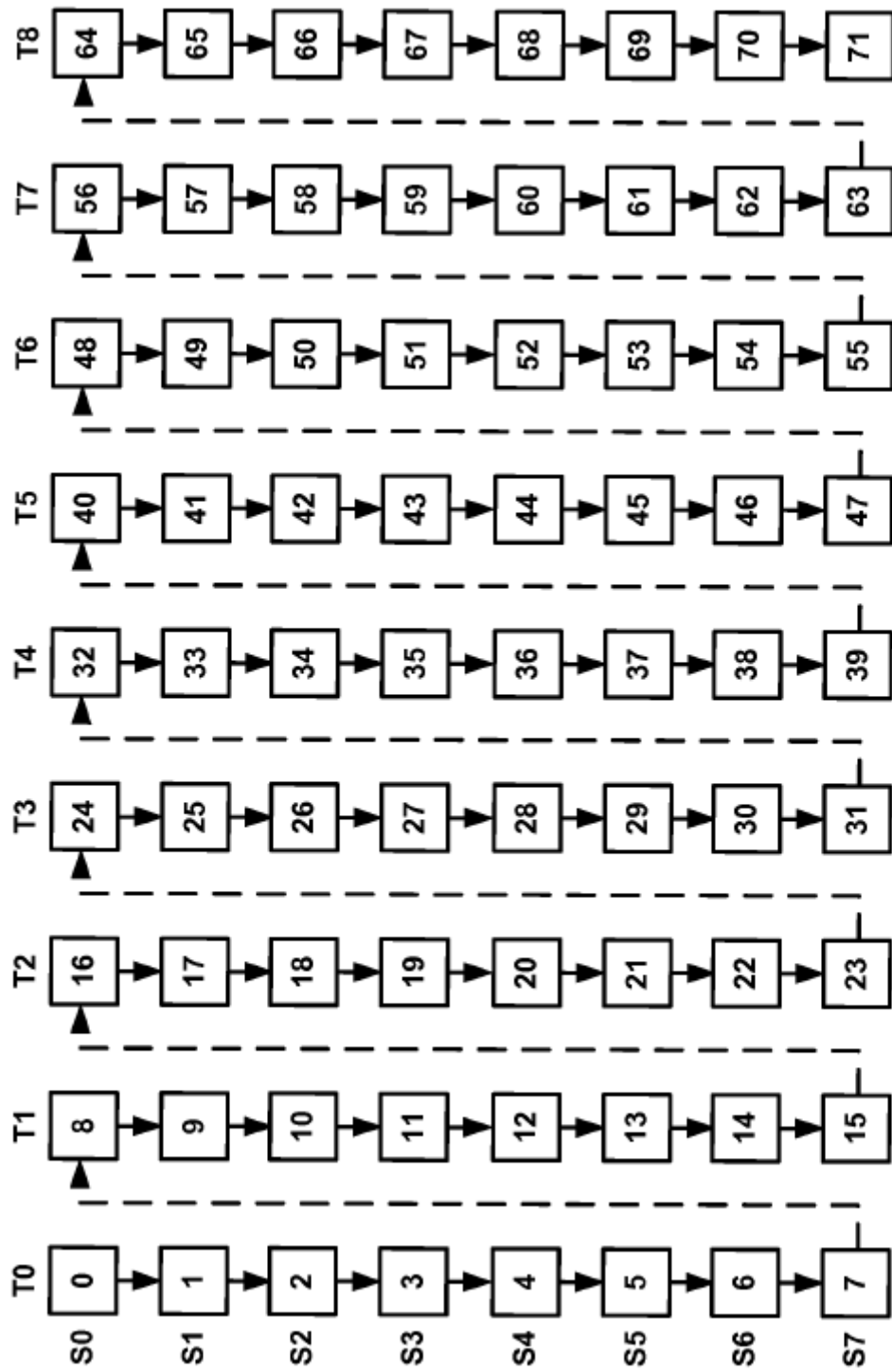
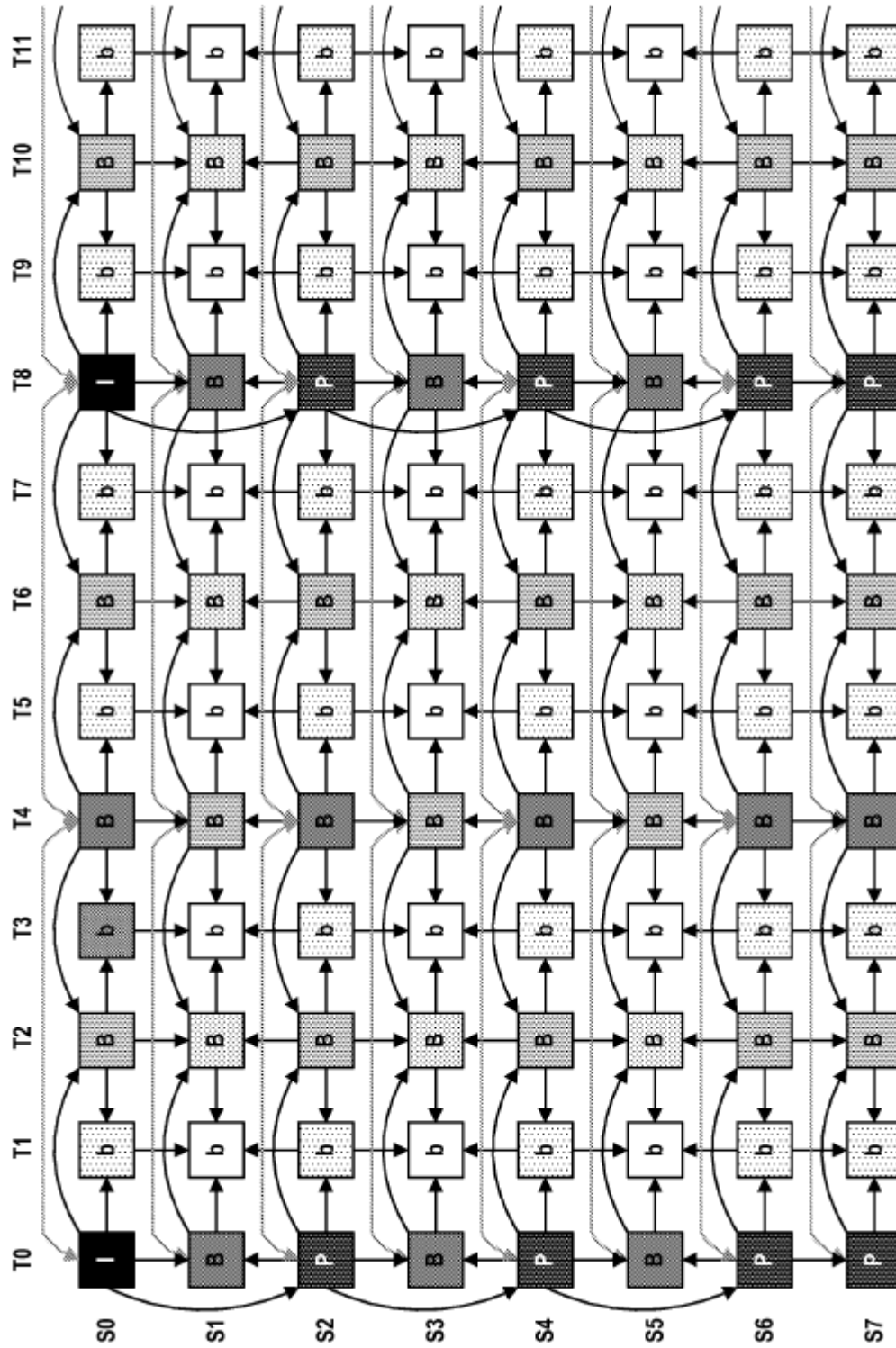


FIG. 2



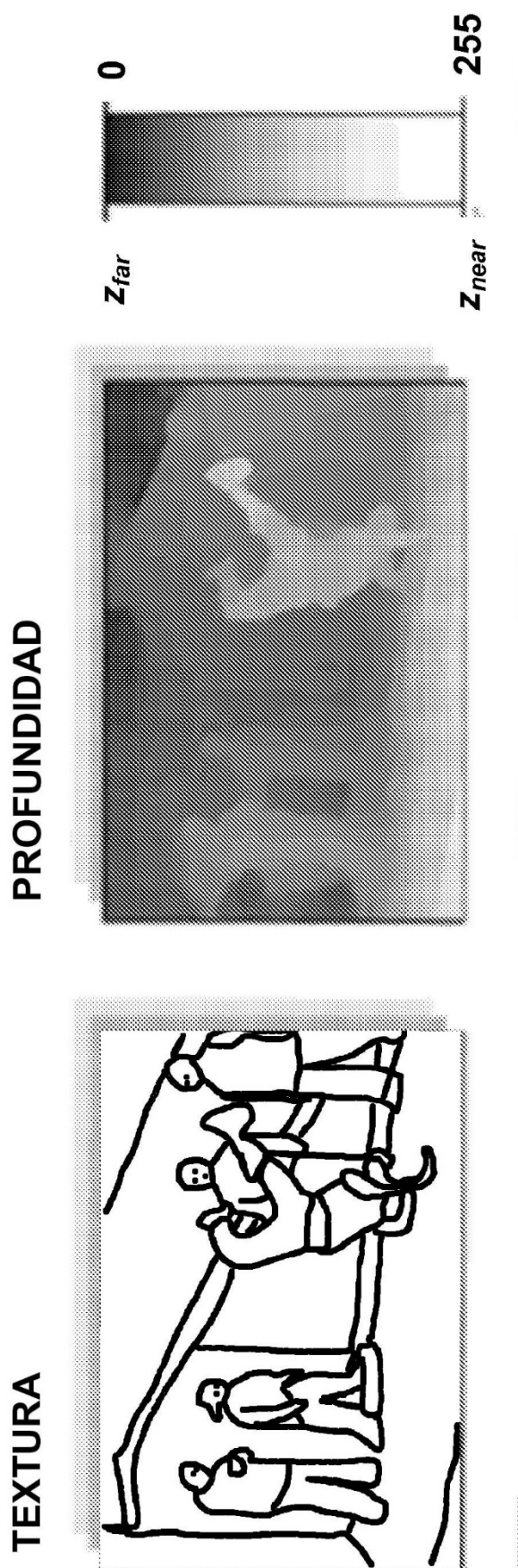


FIG. 4

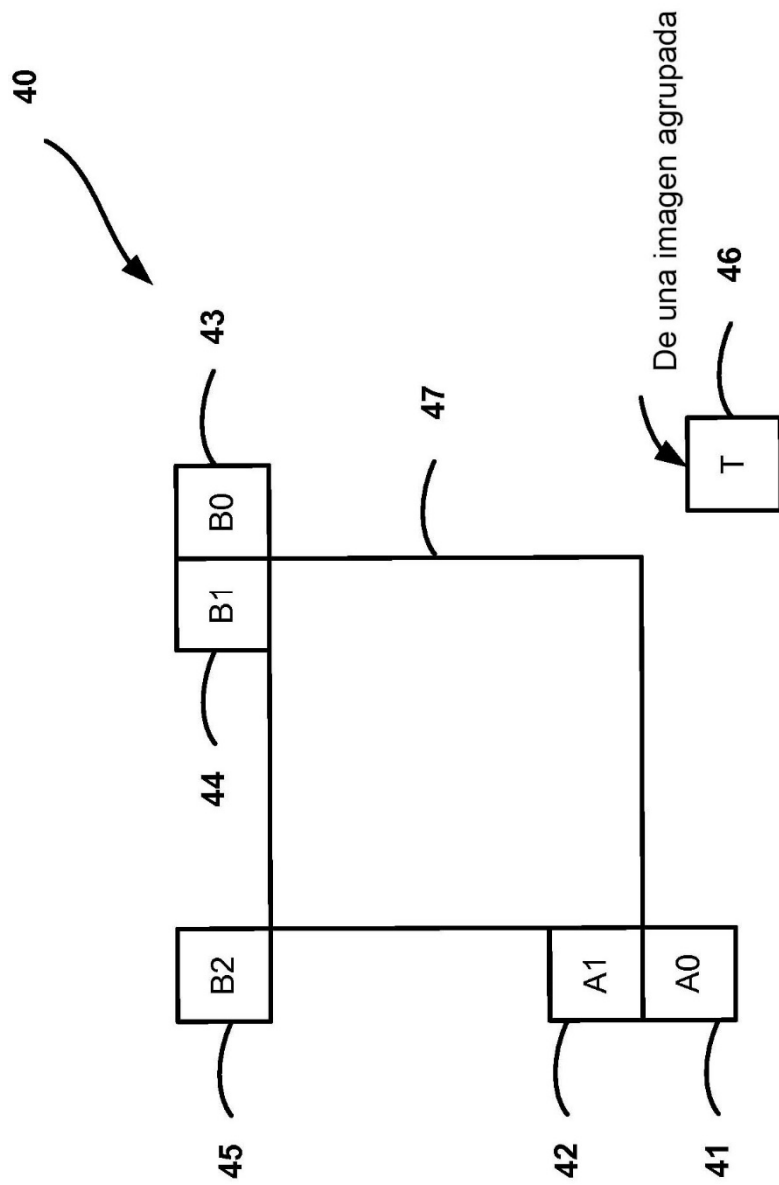


FIG. 5

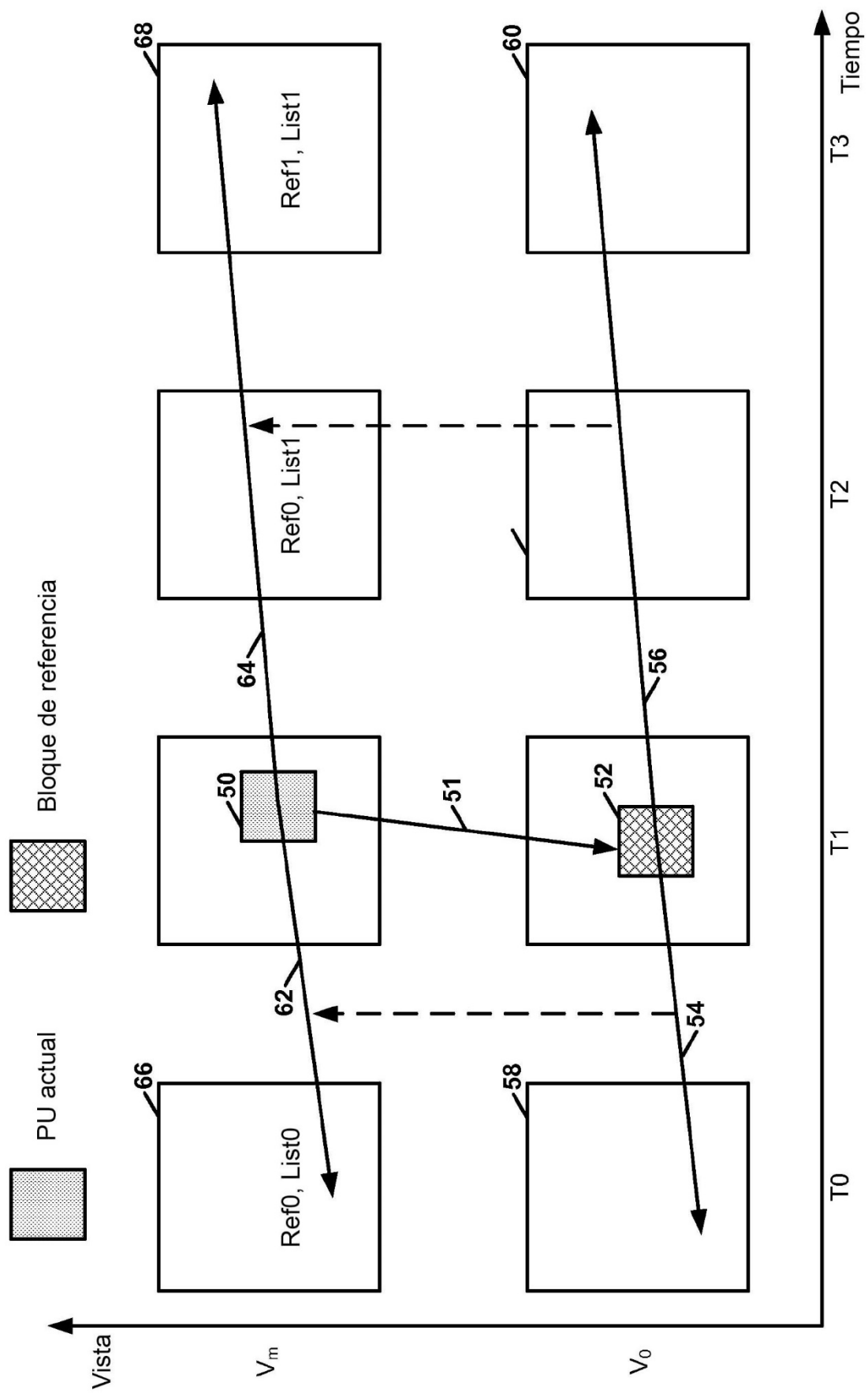


FIG. 6

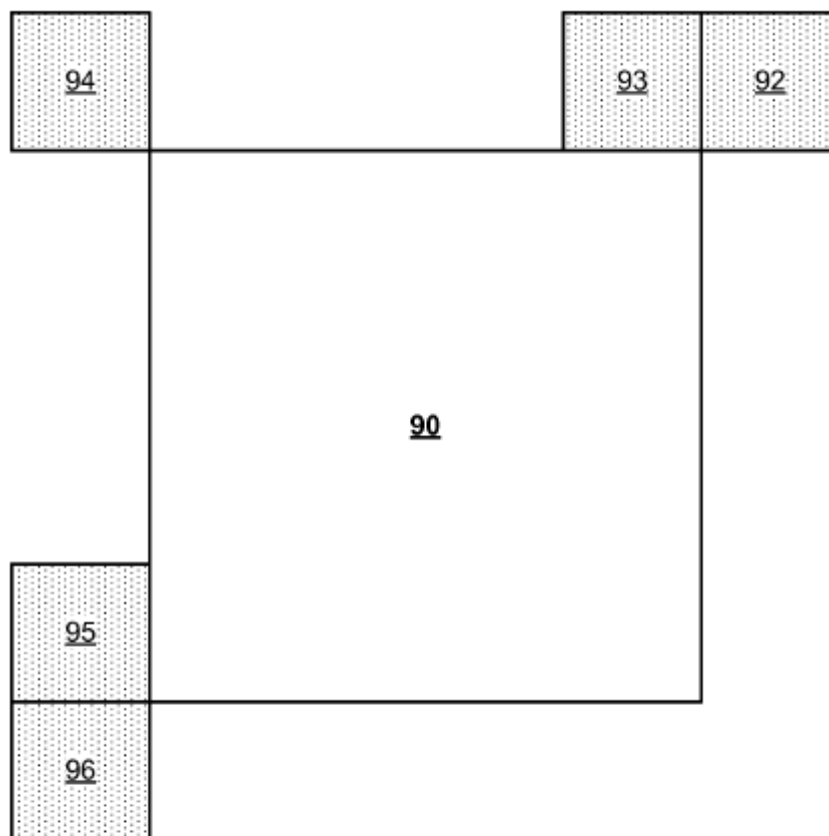


FIG. 7

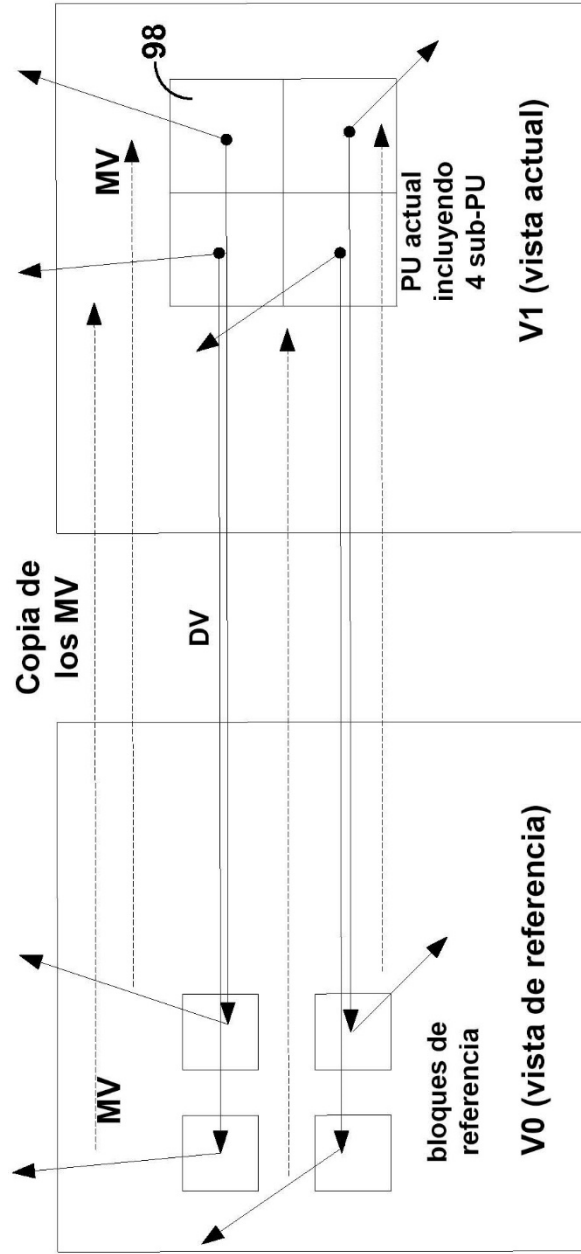


FIG. 8

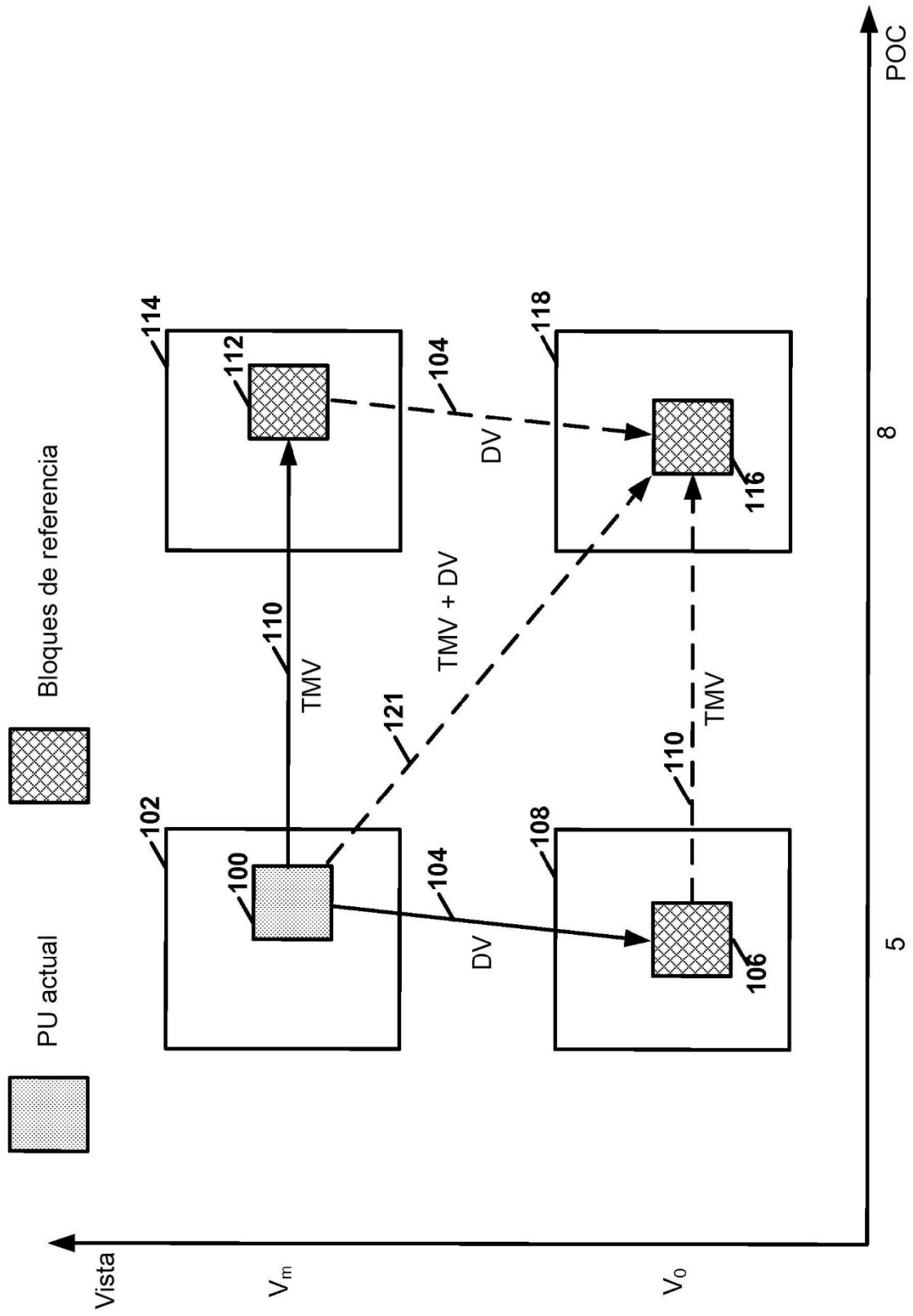


FIG. 9

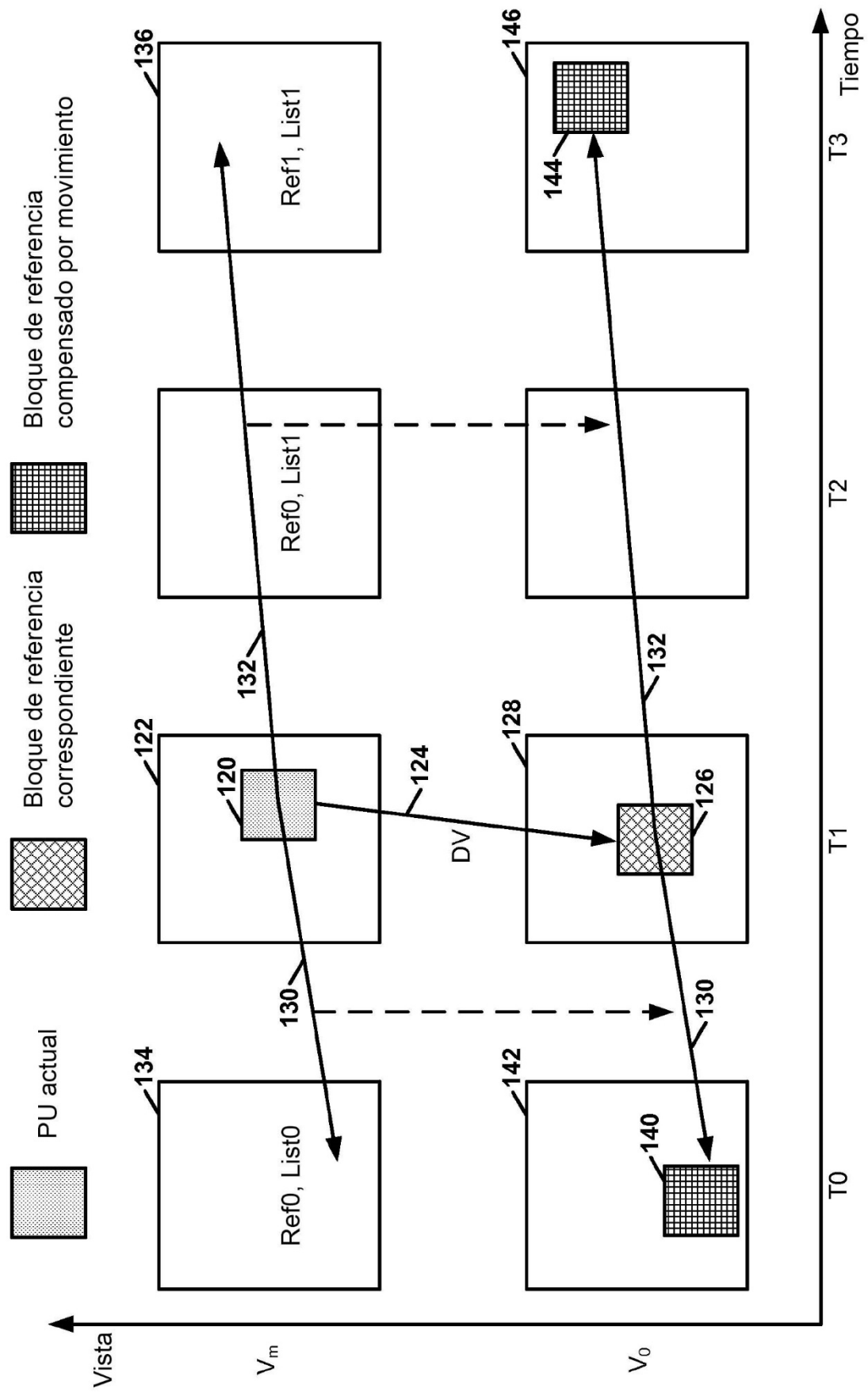


FIG. 10

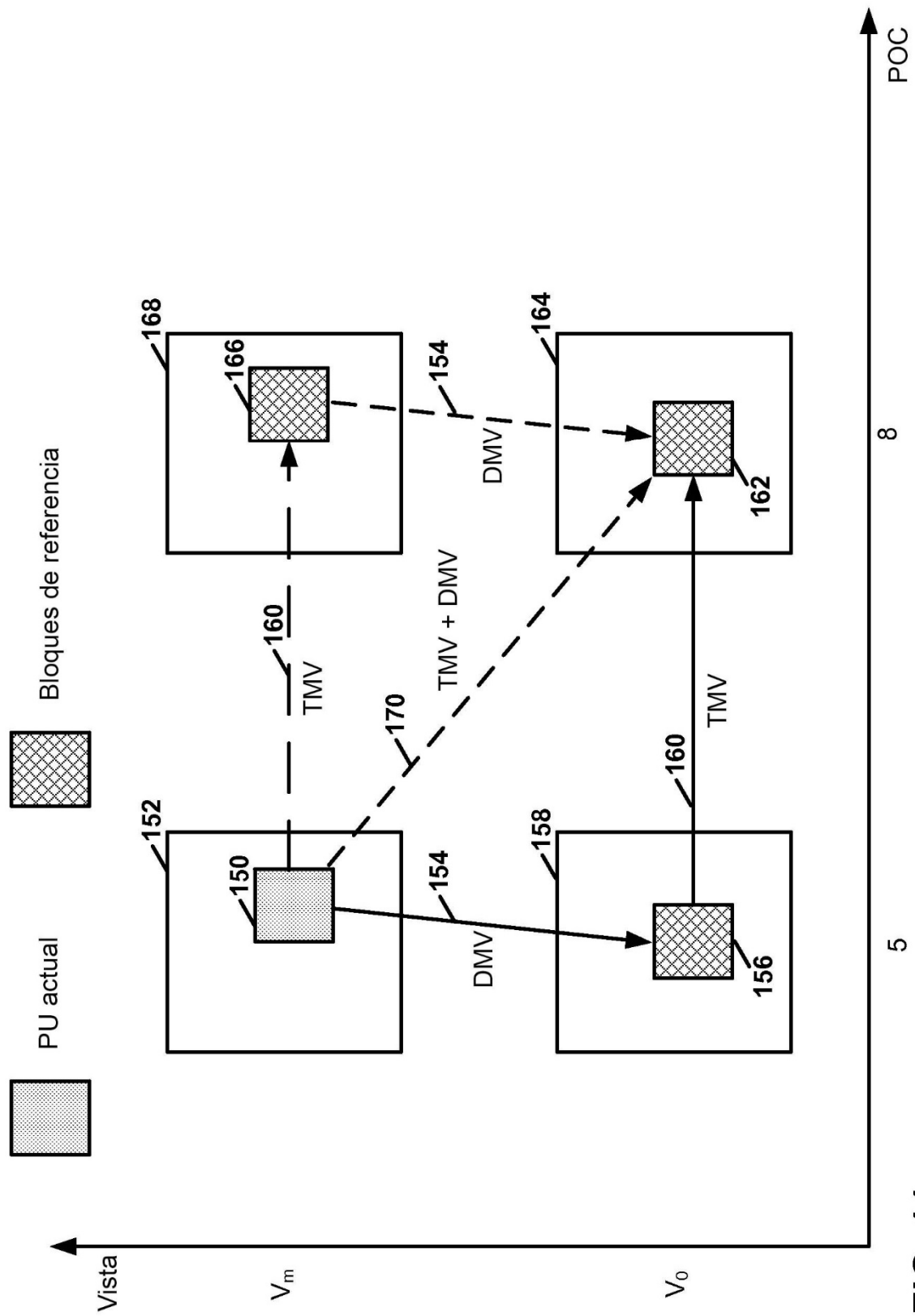


FIG. 11

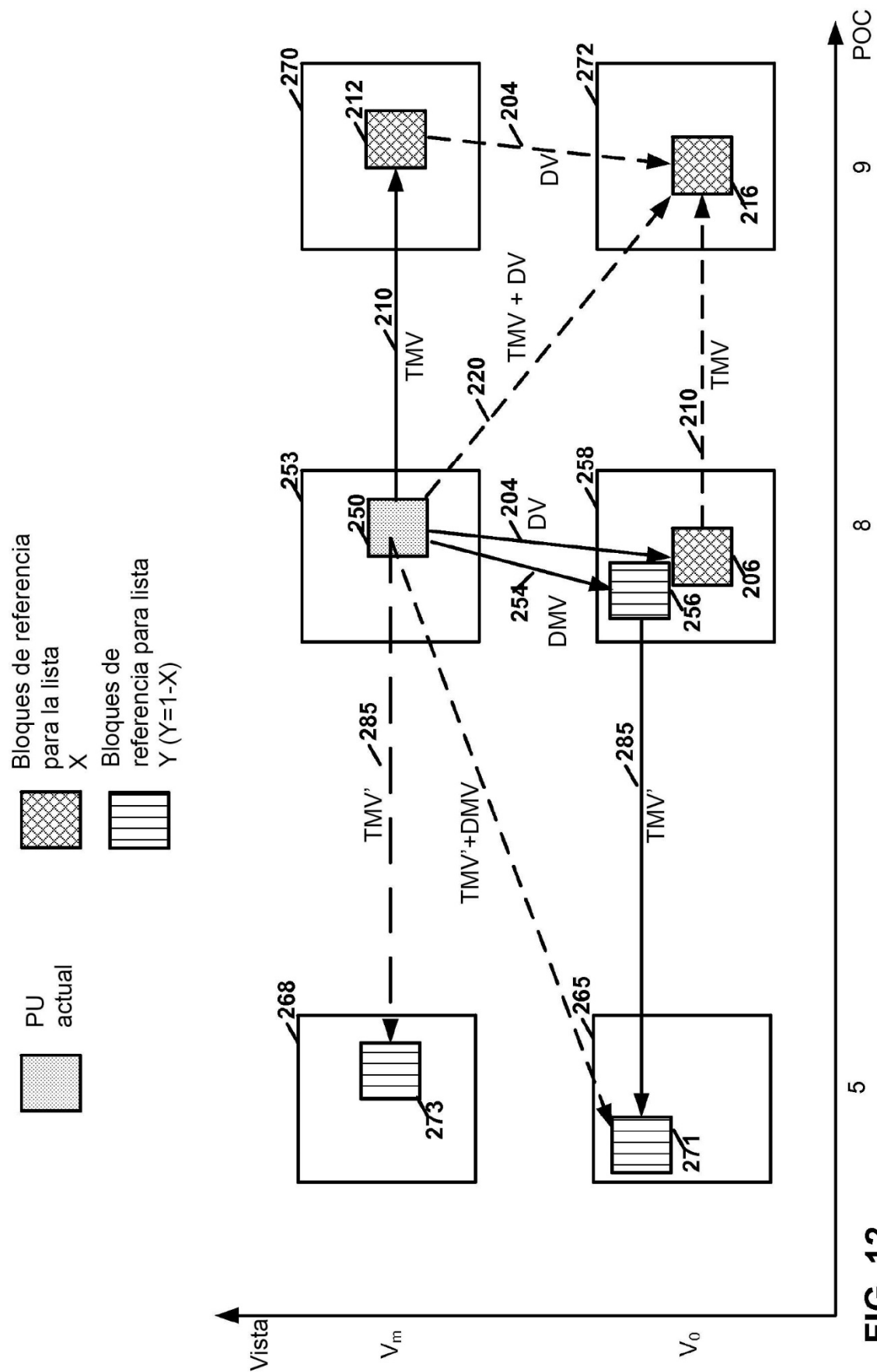


FIG. 12

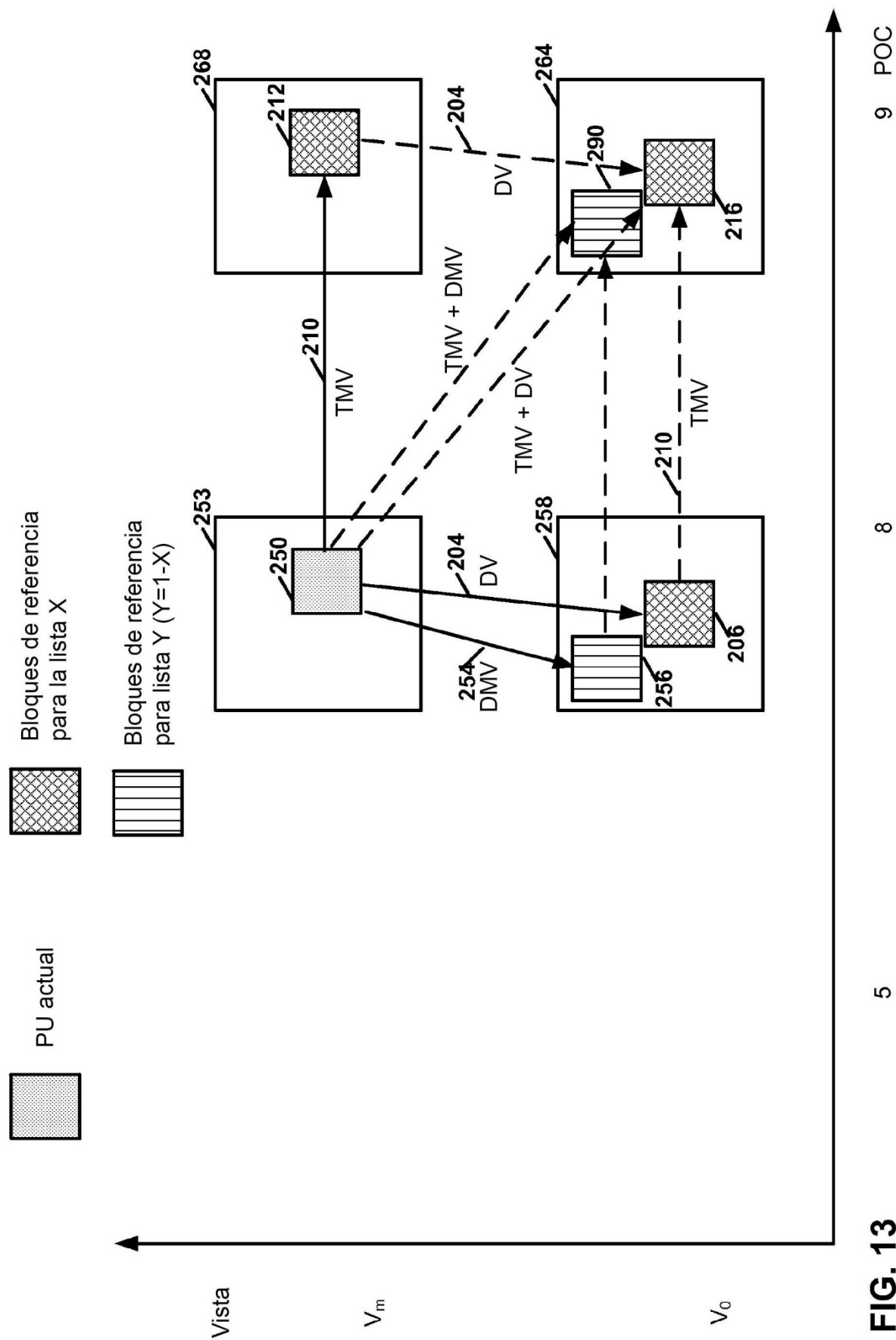


FIG. 13

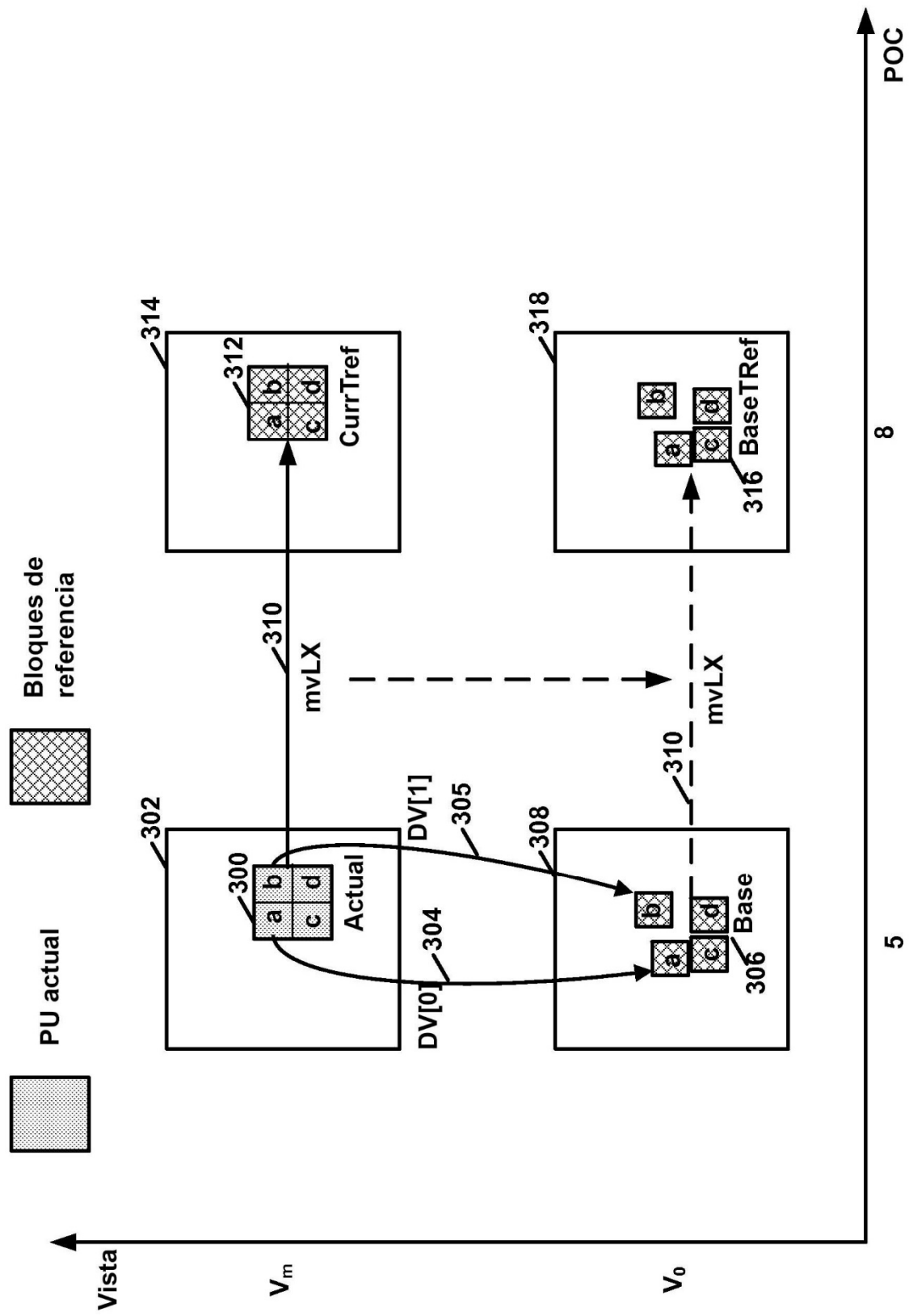


FIG. 14

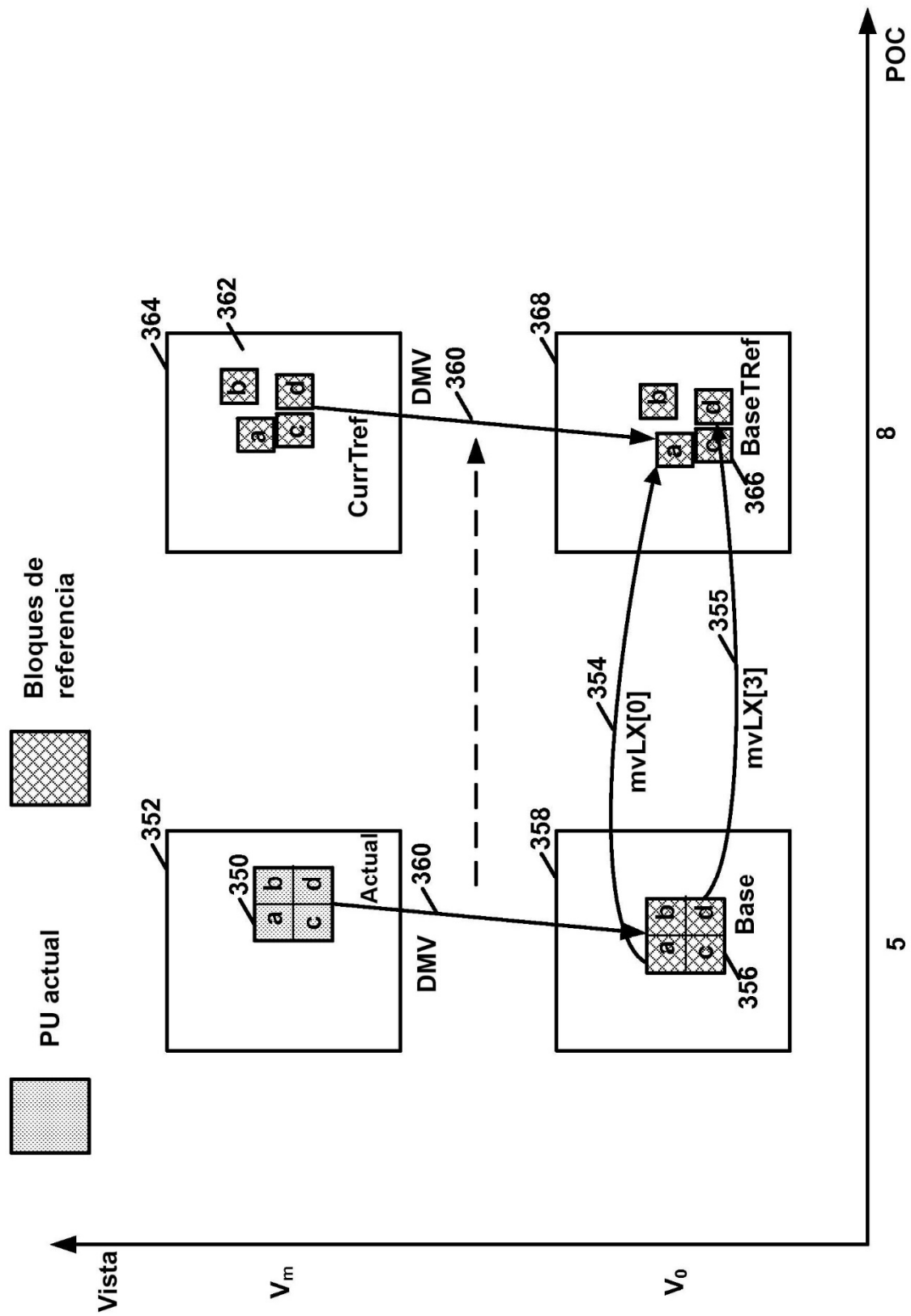


FIG. 15

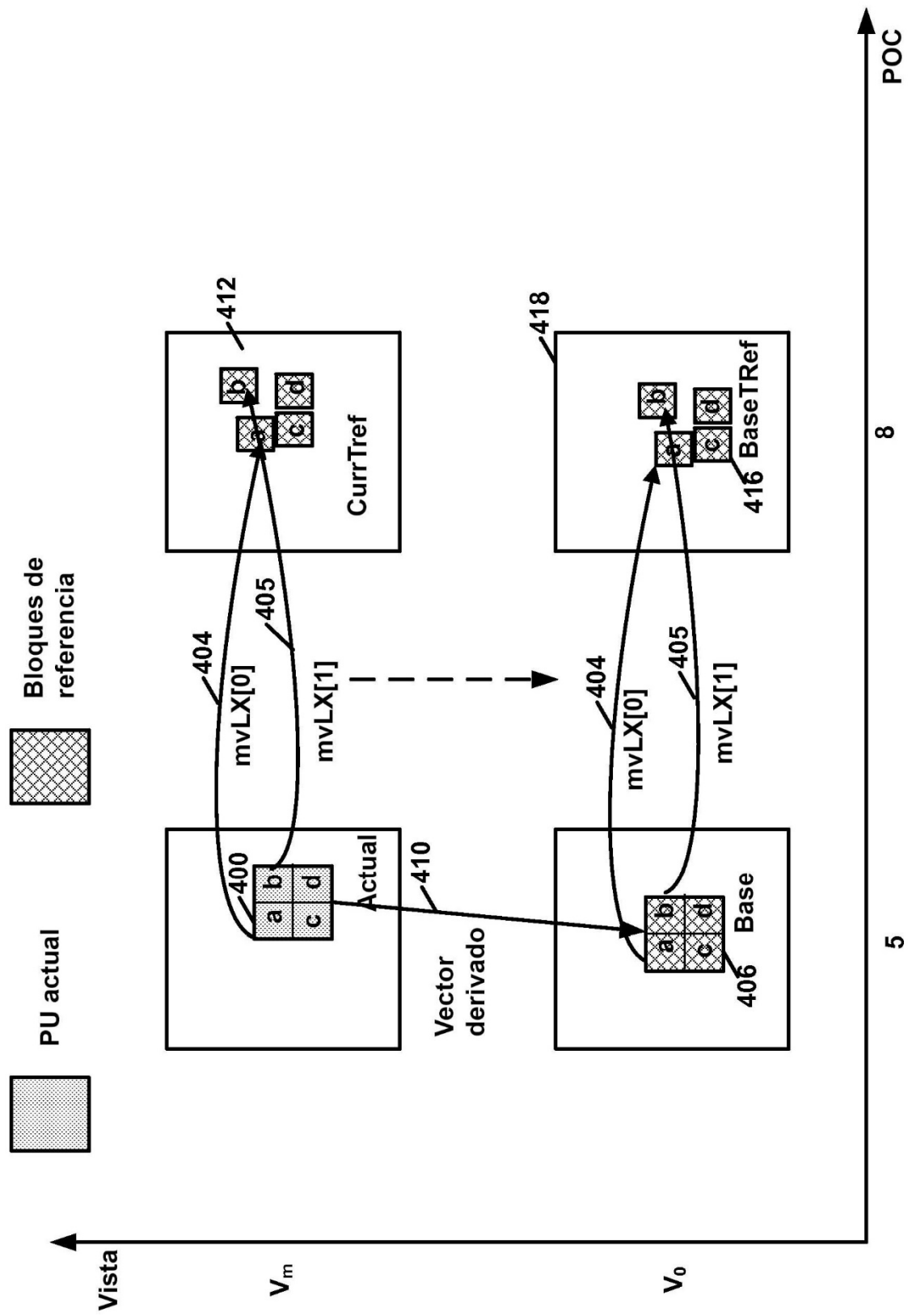


FIG. 16

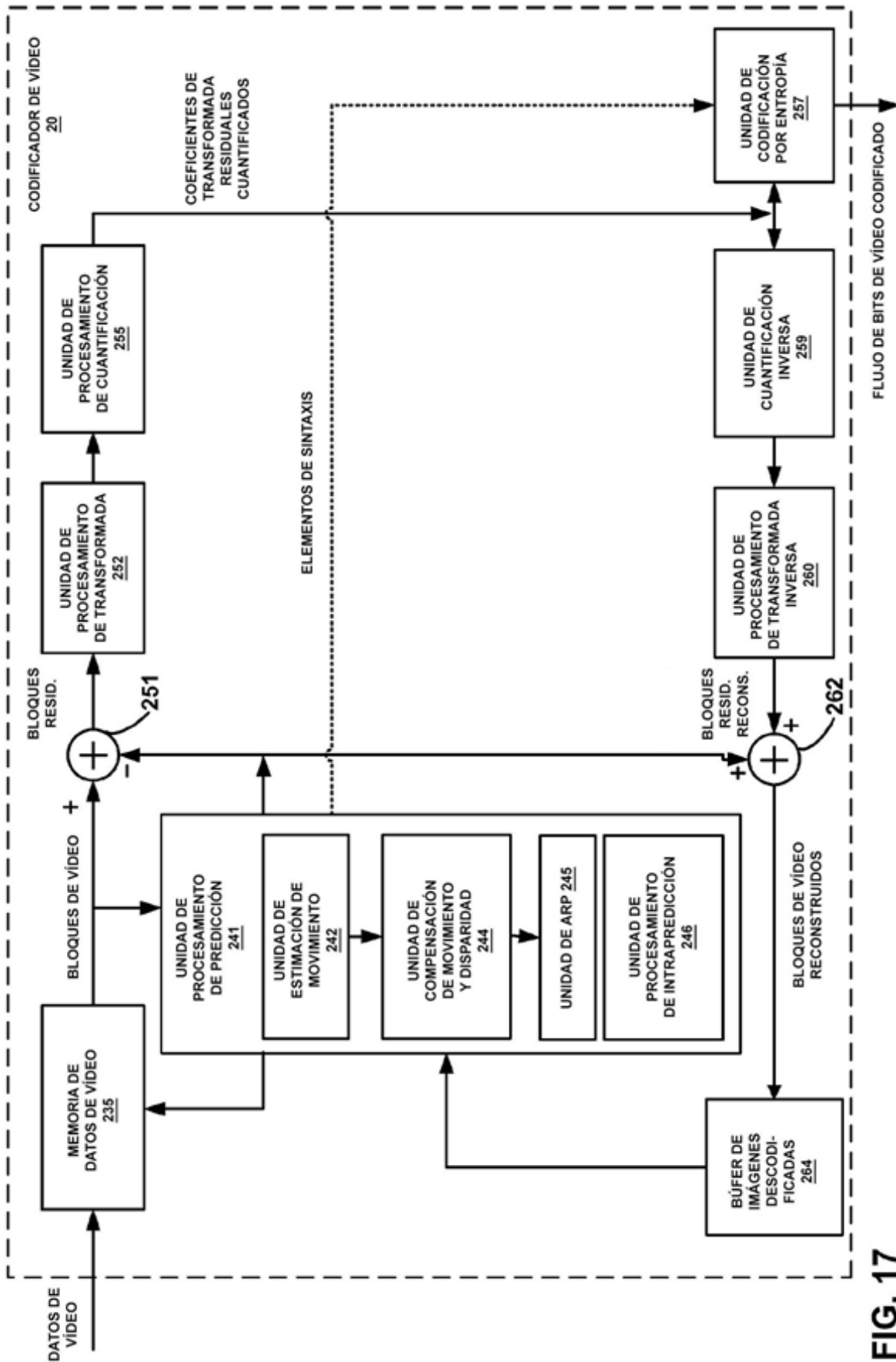


FIG. 17

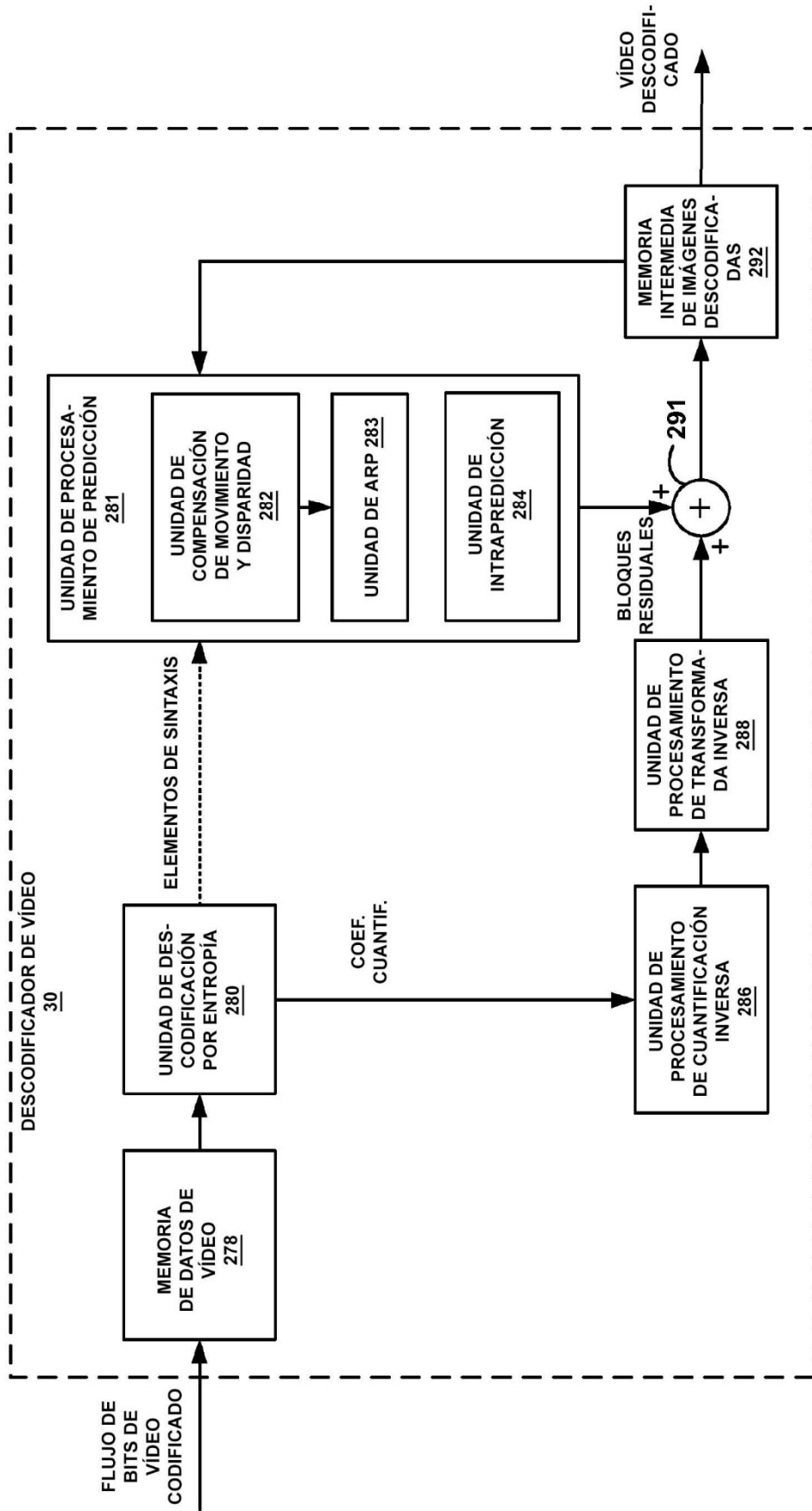


FIG. 18

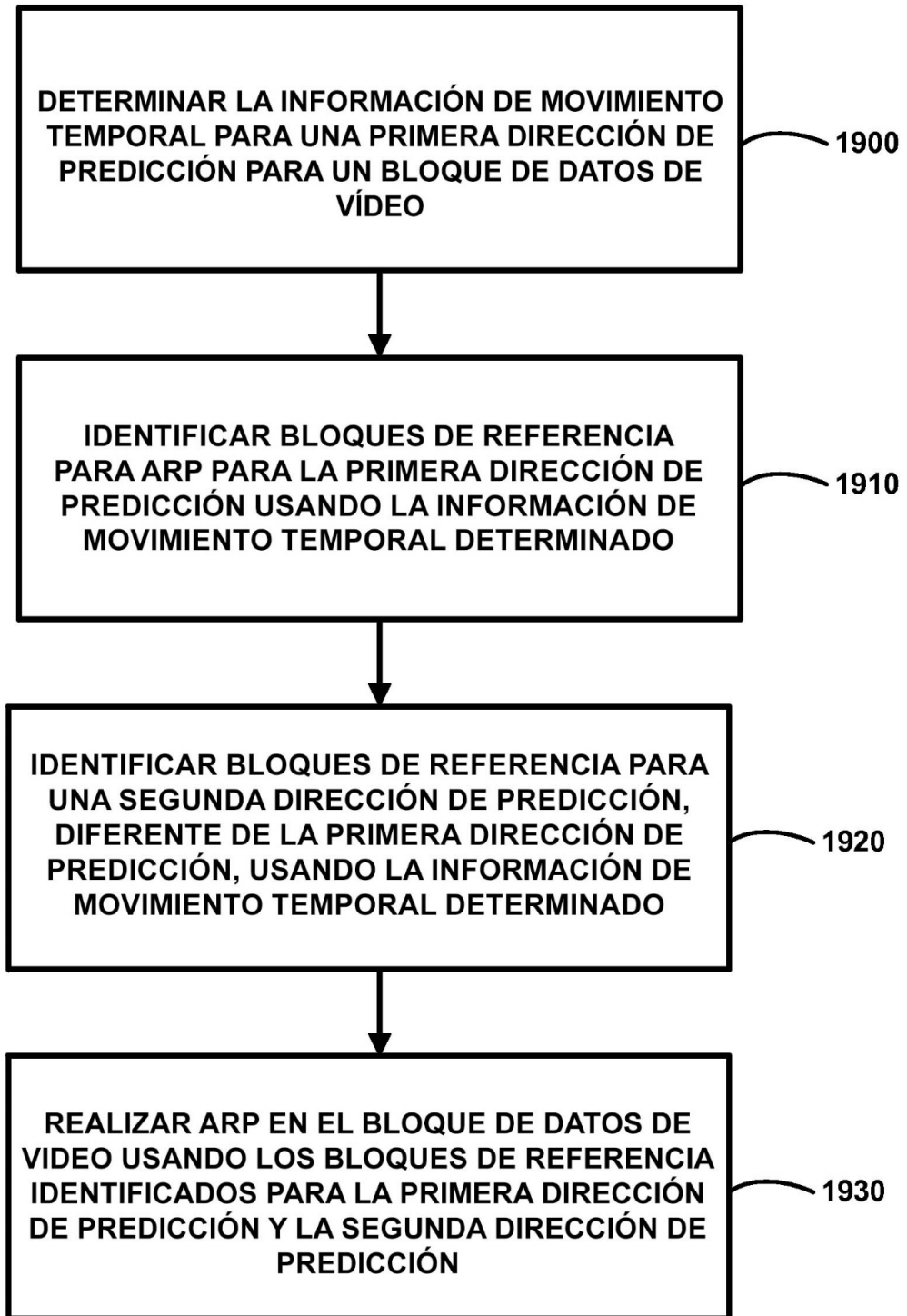


FIG. 19

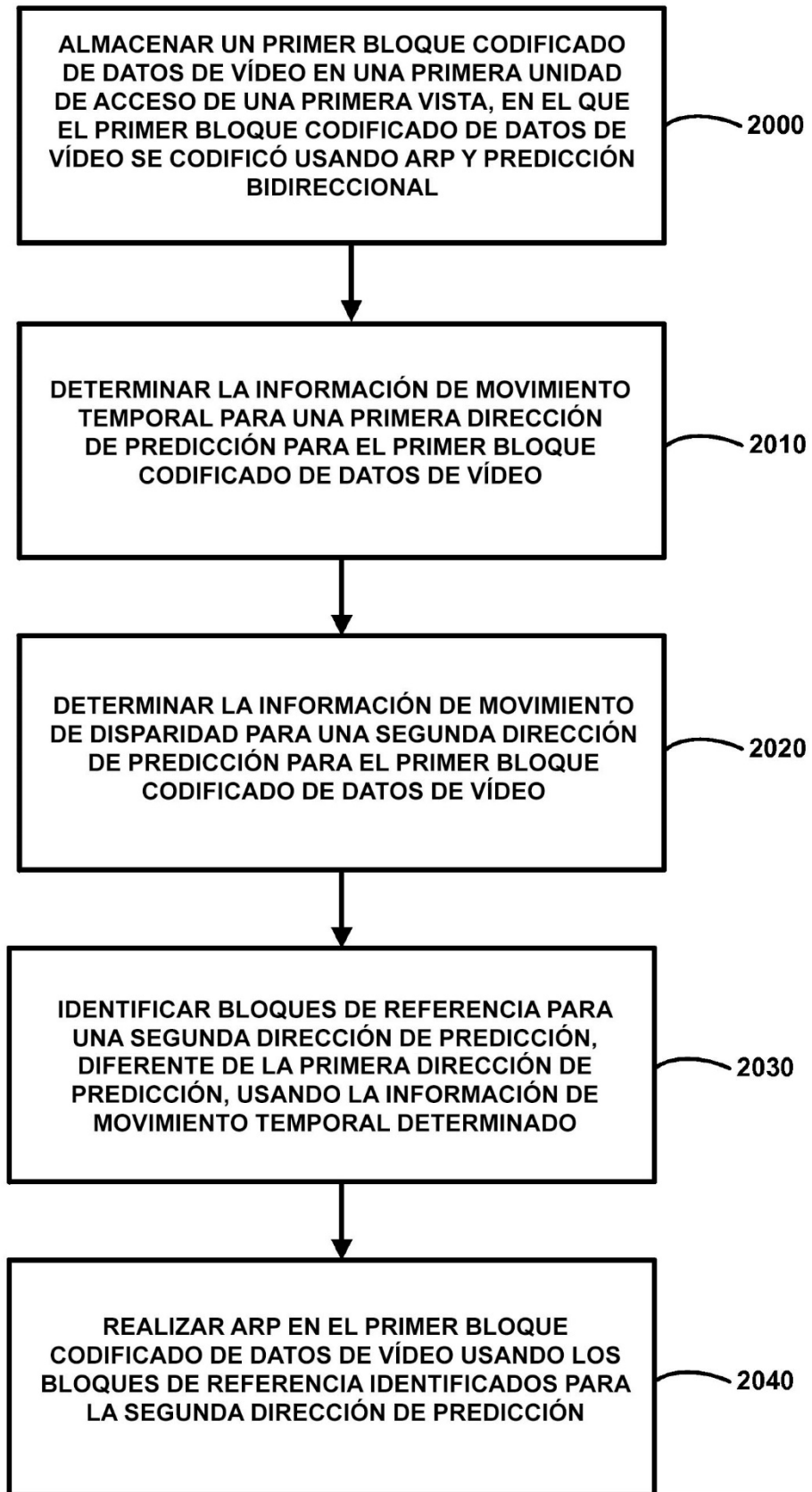


FIG. 20