

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 716 676**

51 Int. Cl.:

H04N 19/597 (2014.01)
H04N 19/105 (2014.01)
H04N 19/176 (2014.01)
H04N 19/513 (2014.01)
H04N 19/186 (2014.01)
H04N 19/577 (2014.01)
H04N 13/00 (2008.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **13.03.2014** **PCT/CN2014/073353**
87 Fecha y número de publicación internacional: **17.09.2015** **WO15135172**
96 Fecha de presentación y número de la solicitud europea: **13.03.2014** **E 14885463 (1)**
97 Fecha y número de publicación de la concesión europea: **26.12.2018** **EP 3117606**

54 Título: **Predicción residual avanzada simplificada para la 3D-HEVC**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
14.06.2019

73 Titular/es:
QUALCOMM INCORPORATED (100.0%)
5775 Morehouse Drive
San Diego, CA 92121-1714, US

72 Inventor/es:
LIU, HONGBIN y
CHEN, YING

74 Agente/Representante:
FORTEA LAGUNA, Juan José

ES 2 716 676 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 simplificada para la 3D-HEVC.

5 CAMPO TÉCNICO

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

ANTECEDENTES

10 **[0002]** Las capacidades de vídeo digital pueden incorporarse a una amplia gama de dispositivos, incluidos televisores digitales, sistemas de difusión directa digital, sistemas de difusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de sobremesa, cámaras digitales, dispositivos de grabación digitales, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos celulares o de radio por satélite, dispositivos de videoconferencia y similares. Los dispositivos de vídeo digital implementan técnicas de compresión de vídeo, tales como las descritas en las normas definidas por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificación Avanzada de Vídeo (AVC), la norma de Codificación de Vídeo de Alta Eficacia (HEVC), actualmente en desarrollo, y las extensiones de dichas normas, para transmitir, recibir y almacenar información de vídeo digital de forma más eficaz.

20 **[0003]** Las extensiones de algunas de las normas mencionadas anteriormente, incluyendo la H.264 / AVC, pueden proporcionar técnicas para la codificación de vídeo de vista múltiple para producir vídeo estéreo o tridimensional ("3D"). En particular, se han propuesto técnicas para la codificación de vista múltiple para su uso en la AVC, con la norma de codificación de vídeo ajustable a escala (SVC) (que es la ampliación ajustable a escala para la H.264/AVC), y la norma de codificación de vídeo de vista múltiple (MVC) (que se ha convertido en la ampliación de vista múltiple para la H.264/AVC).

30 **[0004]** Habitualmente, el vídeo estéreo se logra usando dos vistas, por ejemplo, una vista izquierda y una vista derecha. Una imagen de la vista izquierda se puede exhibir de forma esencialmente simultánea con una imagen de la vista derecha para lograr un efecto de vídeo tridimensional. Por ejemplo, un usuario puede usar gafas pasivas polarizadas que filtran la vista izquierda de la vista derecha. De forma alternativa, las imágenes de las dos vistas se pueden mostrar en rápida sucesión, y el usuario puede usar gafas activas con obturación rápida de los ojos izquierdo y derecho a la misma frecuencia, pero con un desplazamiento de fase de 90 grados.

35 SUMARIO

[0005] En general, esta divulgación describe técnicas para la codificación de vídeo tridimensional. En particular, esta divulgación está relacionada con técnicas para la predicción residual avanzada (ARP) en la 3D-HEVC.

40 **[0006]** En un ejemplo, un procedimiento para codificar datos de vídeo incluye determinar que un bloque actual de una primera vista se codifica utilizando una modalidad de predicción residual avanzada (ARP), en la que el bloque actual se predice bidireccionalmente; determinar un primer vector de movimiento de disparidad y un segundo vector de movimiento de disparidad para el bloque actual; ubicar, con el primer vector de movimiento de disparidad, un primer bloque correspondiente para el bloque actual en una segunda vista; ubicar, con el segundo vector de movimiento de disparidad, un segundo bloque correspondiente del bloque actual en una segunda vista; determinar un vector de movimiento a partir de información de movimiento de al menos uno entre el primer bloque correspondiente del bloque actual y el segundo bloque correspondiente del bloque actual; usar el vector de movimiento, identificando un bloque de referencia del bloque actual en la primera vista, un bloque de referencia del primer bloque correspondiente en la segunda vista y un bloque de referencia del segundo bloque correspondiente en la segunda vista; generar un primer bloque predictivo basándose en el primer bloque correspondiente, el bloque de referencia del bloque actual y el bloque de referencia del primer bloque correspondiente; y generar un segundo bloque predictivo basándose en el segundo bloque correspondiente, el bloque de referencia del bloque actual y el bloque de referencia del segundo bloque correspondiente.

55 **[0007]** En otro ejemplo, un medio de almacenamiento legible por ordenador almacena en el mismo instrucciones que, cuando se ejecutan, hacen que uno o más procesadores realicen los procedimientos o la combinación de procedimientos anteriores.

60 **[0008]** En otro ejemplo, un dispositivo para codificar vídeo incluye un codificador de vídeo configurado para determinar que un bloque actual de una primera vista se codifica utilizando una modalidad de predicción residual avanzada (ARP), en la que el bloque actual se predice bidireccionalmente; determinar un primer vector de movimiento de disparidad y un segundo vector de movimiento de disparidad para el bloque actual; ubicar, con el primer vector de movimiento de disparidad, un primer bloque correspondiente para el bloque actual en una segunda vista; ubicar, con el segundo vector de movimiento de disparidad, un segundo bloque correspondiente del bloque actual en una segunda vista; determinar un vector de movimiento a partir de información de movimiento de al menos uno entre el primer bloque correspondiente del bloque actual y el segundo bloque correspondiente del bloque actual; utilizando el vector

de movimiento, identificar un bloque de referencia del bloque actual en la primera vista, un bloque de referencia del primer bloque correspondiente en la segunda vista y un bloque de referencia del segundo bloque correspondiente en la segunda vista; generar un primer bloque predictivo basándose en el primer bloque correspondiente, el bloque de referencia del bloque actual y el bloque de referencia del primer bloque correspondiente; generar un segundo bloque predictivo basándose en el segundo bloque correspondiente, el bloque de referencia del bloque actual y el bloque de referencia del segundo bloque correspondiente.

[0009] Los detalles de uno o más ejemplos se exponen en los dibujos adjuntos y en la siguiente descripción. Otras características, objetos y ventajas resultarán evidentes a partir de la descripción y de los dibujos, y a partir de las reivindicaciones.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

[0010]

- La figura 1 es un diagrama de bloques que ilustra un sistema ejemplar de codificación y decodificación de vídeo que puede utilizar las técnicas descritas en esta divulgación.
- La figura 2 es un diagrama conceptual que ilustra un orden ejemplar de decodificación de vista múltiple.
- La figura 3 es un diagrama conceptual que ilustra una estructura ejemplar de predicción para la codificación de vista múltiple.
- La figura 4 muestra un ejemplo de bloques en vecindad espacial con respecto a una unidad de codificación.
- La figura 5 muestra una estructura ejemplar de predicción para la predicción residual avanzada (ARP) en la 3D-HEVC.
- La figura 6 muestra una relación ejemplar entre un bloque actual, un bloque de referencia y un bloque compensado por movimiento en la codificación de vídeo de vista múltiple.
- La figura 7 muestra un ejemplo de la ARP para datos residuales entre vistas.
- La figura 8A muestra una estructura ejemplar de predicción de la ARP temporal a nivel de bloque.
- La figura 8B muestra una estructura ejemplar de predicción de la ARP entre vistas a nivel de bloque.
- La figura 9 muestra una estructura ejemplar de predicción para la predicción de movimiento entre vistas a nivel de sub-PU.
- La figura 10A muestra una estructura ejemplar de predicción de la ARP temporal a nivel de sub-PU.
- La figura 10B muestra una estructura ejemplar de predicción de la ARP entre vistas a nivel de sub-PU.
- La figura 11 muestra un ejemplo de acceso de bloques de referencia para la ARP bidireccional entre vistas en la 3D-HEVC.
- La figura 12 muestra un acceso ejemplar a bloques de referencia para la ARP temporal y la ARP entre vistas en la 3D-HEVC.
- La figura 13 muestra un ejemplo de un único vector de movimiento temporal utilizado en la ARP bidireccional entre vistas.
- La figura 14 muestra un ejemplo de un codificador de vídeo configurado para implementar las técnicas descritas en esta divulgación.
- La figura 15 muestra un ejemplo de un decodificador de vídeo configurado para implementar las técnicas descritas en esta divulgación.
- La figura 16 muestra un procedimiento ejemplar de predicción de un bloque de vídeo de acuerdo a las técnicas de esta divulgación.
- La figura 17 muestra un procedimiento de predicción de un bloque de vídeo.
- La figura 18 muestra un procedimiento de predicción de un bloque de vídeo.

DESCRIPCIÓN DETALLADA

[0011] Esta divulgación introduce técnicas relacionadas con la predicción residual avanzada (ARP) para la 3D-HEVC. Las técnicas de esta divulgación pueden ser realizadas por un codificador de vídeo, tal como un codificador de vídeo o un decodificador de vídeo. En ARP, un codificador de vídeo genera un predictor residual basándose en una diferencia entre imágenes ya codificadas. El codificador de vídeo luego agrega este predictor residual a un bloque predictivo original para generar un bloque predictivo final. El bloque predictivo final, que incluye el predictor residual, es potencialmente un mejor predictor, es decir, se asemeja más estrechamente al bloque predicho que el predictor original.

[0012] En general, existen dos tipos de ARP, a las que se hace referencia en esta divulgación como ARP temporal y ARP entre vistas. En la ARP temporal, para un bloque actual en una primera vista, un codificador de vídeo localiza un bloque correspondiente en una segunda vista usando un vector de disparidad para el bloque actual. En esta divulgación, este bloque correspondiente en la segunda vista se denominará bloque base. Usando un vector de movimiento temporal del bloque actual, un codificador de vídeo localiza un bloque de referencia del bloque actual en una imagen diferente de la primera vista. En esta divulgación, este bloque se menciona como bloque de referencia actual. Usando el mismo vector de movimiento temporal utilizado para identificar el bloque de referencia actual, un codificador de vídeo localiza un bloque de referencia del bloque base en una imagen de la segunda vista. En esta divulgación, este bloque se mencionará como bloque base de referencia. La diferencia entre el bloque base y el bloque base de referencia se puede calcular como un predictor residual. El codificador de vídeo luego agrega el predictor residual, posiblemente con un factor de ponderación, al bloque de referencia actual para determinar un predictor final.

[0013] En la ARP entre vistas, para un bloque actual en una primera vista, un codificador de vídeo localiza un bloque correspondiente en una segunda vista usando un vector de movimiento de disparidad para el bloque actual. Usando un vector de movimiento temporal del bloque base, el codificador de vídeo localiza un bloque base de referencia del bloque base en una imagen diferente de la segunda vista. Utilizando el mismo vector de movimiento temporal utilizado para identificar el bloque de referencia base, el codificador de vídeo identifica un bloque de referencia actual del bloque actual en una imagen de la primera vista. El codificador de vídeo calcula la diferencia entre el bloque de referencia actual y el bloque de referencia de base y utiliza la diferencia calculada como un predictor residual. El codificador de vídeo luego agrega este predictor residual, posiblemente con un factor de ponderación, al bloque base para determinar un predictor final.

[0014] Cuando un codificador de vídeo codifica un bloque predicho bidireccionalmente usando la ARP, el codificador de vídeo debe evaluar bloques de referencia adicionales para las dos direcciones de predicción, lo que aumenta la complejidad general. Cuando un codificador de vídeo codifica un bloque usando la ARP, la ARP puede usarse para codificar tanto los componentes cromáticos del bloque como los componentes de la luminancia del bloque, lo que aumenta aún más la complejidad general. Esta divulgación introduce varias simplificaciones potenciales a las técnicas conocidas de la ARP. En un ejemplo, de acuerdo a las técnicas de esta divulgación, cuando se realiza la ARP entre vistas para un bloque predicho bidireccionalmente, el codificador de vídeo puede determinar un vector de movimiento para un primer bloque correspondiente como parte de la realización de la ARP para una primera dirección de predicción y reutilizar ese vector de movimiento determinado al realizar la ARP para una segunda dirección de predicción. De acuerdo a otro ejemplo, para un bloque predicho bidireccionalmente, un codificador de vídeo puede aplicar la ARP en solamente una dirección para un componente de croma de un bloque, pero aplicar la ARP en dos direcciones para un componente de luma del bloque. De acuerdo a otro ejemplo, un codificador de vídeo puede aplicar selectivamente la ARP a los componentes de croma basándose en el tamaño del bloque. Estas simplificaciones, así como otras técnicas incluidas en esta divulgación, pueden reducir la complejidad general de la codificación.

[0015] La figura 1 es un diagrama de bloques que ilustra un sistema ejemplar de codificación y decodificación de vídeo 10 que puede configurarse para realizar las técnicas de ARP descritas en esta divulgación. Como se muestra en la figura 1, el sistema 10 incluye un dispositivo de origen 12 que genera datos de vídeo codificados, a decodificar en un momento posterior por parte de un dispositivo de destino 14. El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender cualquiera entre una amplia gama de dispositivos, incluyendo ordenadores de sobremesa, ordenadores plegables (es decir, portátiles), ordenadores de tipo tableta, decodificadores, equipos telefónicos tales como los denominados teléfonos "inteligentes", los denominados paneles "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de transmisión 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.

[0016] El sistema 10 puede funcionar de acuerdo a diferentes normas de codificación de vídeo, una norma patentada o cualquier otra forma de codificación de vídeo múltiple. A continuación se describen unos pocos ejemplos de normas de codificación de vídeo, y no se deben considerar limitantes. Entre las normas de codificación de vídeo se incluyen ITU-T H.261, MPEG-1 Visual de la ISO/IEC, ITU-T H.262 o MPEG-2 Visual de la ISO/IEC, ITU-T H.263, MPEG-4 Visual de la ISO/IEC e ITU-T H.264 (también conocida como AVC del MPEG-4 de la ISO/IEC), incluyendo sus ampliaciones de codificación de vídeo ajustable a escala (SVC) y de codificación de vídeo de múltiples vistas (MVC). El más reciente borrador conjunto de la MVC se describe en "Advanced vídeo coding for generic audiovisual services" ["Codificación de vídeo avanzada para servicios audiovisuales genéricos"], Recomendación H.264 de la ITU-T, marzo

de 2010. Otro borrador conjunto de la MVC se describe en "Advanced video coding for generic audiovisual services [Codificación de vídeo avanzada para servicios audiovisuales genéricos]", Recomendación ITU-T H.264, junio de 2011. Algunas normas adicionales de codificación de vídeo incluyen la MVC+D y la 3D-AVC, que se basan en la AVC. Además, una nueva norma de codificación de vídeo, concretamente, la Codificación de Vídeo de Alta Eficacia (HEVC), ha sido desarrollada por el Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) del Grupo de Expertos en Codificación de Vídeo (VCEG) de la ITU-T y el Grupo de Expertos en Imágenes en Movimiento (MPEG) de ISO/IEC.

[0017] Con fines de ilustración solamente, las técnicas descritas en esta divulgación se describen con ejemplos de acuerdo a la norma H.264, tales como la 3D-AVC. Sin embargo, las técnicas descritas en esta divulgación no se deberían considerar limitadas a estas normas ejemplares, y se pueden ampliar a otras normas de codificación de vídeo para la codificación de vista múltiple o la codificación de vídeo tridimensional (por ejemplo, la 3D-HEVC), o a técnicas relacionadas con la codificación de vista múltiple o la codificación de vídeo tridimensional, que no se basan necesariamente en una norma de codificación de vídeo particular. Por ejemplo, las técnicas descritas en esta divulgación se implementan mediante codificadores/decodificadores de vídeo (códecs) para la codificación de vista múltiple, donde la codificación de vista múltiple incluye la codificación de dos o más vistas.

[0018] El dispositivo de destino 14 puede recibir los datos de vídeo codificados a decodificar, mediante un enlace 16. El enlace 16 puede comprender cualquier tipo de medio o dispositivo capaz de desplazar los datos de vídeo codificados desde el dispositivo de origen 12 hasta el dispositivo de destino 14. En un ejemplo, el enlace 16 puede comprender un medio de comunicación para permitir que el dispositivo de origen 12 transmita datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real. Los datos de vídeo codificados se pueden modular de acuerdo a una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitir al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación, inalámbrica o cableada, tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión física. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área amplia o una red global tal como Internet. El medio de comunicación puede incluir encaminadores, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo de origen 12 hasta el dispositivo de destino 14.

[0019] De forma alternativa, los datos codificados pueden emitirse desde la interfaz de salida 22 hasta un dispositivo de almacenamiento 34. De forma similar, una interfaz de entrada puede acceder a los datos codificados procedentes del dispositivo de almacenamiento 34. El dispositivo de almacenamiento 34 puede incluir cualquiera entre varios medios de almacenamiento de datos distribuidos o de acceso local, tales como una unidad de disco duro, los discos Blu-ray, los DVD, los CD-ROM, memoria flash, memoria volátil o no volátil, o cualquier otro medio de almacenamiento digital adecuado para almacenar datos de vídeo codificados. En otro ejemplo, el dispositivo de almacenamiento 34 puede corresponder a un servidor de ficheros o a otro dispositivo de almacenamiento intermedio que pueda contener el vídeo codificado generado por el dispositivo de origen 12. El dispositivo de destino 14 puede acceder a datos de vídeo almacenados procedentes del dispositivo de almacenamiento 34, por medio de transmisión continua o descarga. El servidor de ficheros puede ser cualquier tipo de servidor capaz de almacenar datos de vídeo codificados y transmitir esos datos de vídeo codificados al dispositivo de destino 14. Los servidores ejemplares de ficheros incluyen un servidor de la Red (por ejemplo, para una sede de la Red), un servidor del FTP, dispositivos de almacenamiento conectado 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 de Wi-Fi), una conexión cableada (por ejemplo, DSL, módem de cable, etc.) o una combinación de ambos que sea adecuada para acceder a datos de vídeo codificados almacenados en un servidor de ficheros. La transmisión de datos de vídeo codificados desde el dispositivo de almacenamiento 34 puede ser una transmisión continua, una transmisión de descarga o una combinación de ambas.

[0020] Las técnicas de esta divulgación para la ARP no se limitan necesariamente a aplicaciones o configuraciones inalámbricas. Las técnicas se pueden aplicar a la codificación de vídeo como apoyo a cualquiera entre varias aplicaciones de multimedios, tales como difusiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo en continuo, por ejemplo, mediante Internet, codificación de vídeo digital para su almacenamiento en un medio de almacenamiento de datos, decodificación de vídeo digital almacenado en un medio de almacenamiento de datos u otras aplicaciones. En algunos ejemplos, el sistema 10 se puede configurar para prestar soporte a la transmisión de vídeo unidireccional o bidireccional, para dar soporte a aplicaciones tales como la transmisión de vídeo en continuo, la reproducción de vídeo, la difusión de vídeo y/o la videotelefonía.

[0021] En el ejemplo de la figura 1, el dispositivo de origen 12 incluye un origen de vídeo 18, un codificador de vídeo 20 y una interfaz de salida 22. Como se explicará con mayor detalle a continuación, el codificador de vídeo 20 puede configurarse para realizar las técnicas de ARP descritas en esta divulgación. En algunos casos, la interfaz de salida 22 puede incluir un modulador/demodulador (módem) y/o un transmisor. En el dispositivo de origen 12, el origen de vídeo 18 puede incluir un origen tal como un dispositivo de captura de vídeo, por ejemplo, una videocámara, un archivo de vídeo que contiene vídeo capturado previamente, una interfaz de señales de vídeo para recibir vídeo desde un proveedor de contenido de vídeo y/o un sistema de gráficos por ordenador para generar datos de gráficos por

ordenador como el vídeo de origen, o una combinación de dichos orígenes. En un ejemplo, si el origen de vídeo 18 es una videocámara, el dispositivo de origen 12 y el dispositivo de destino 14 pueden formar los denominados teléfonos con cámara o videoteléfonos. Sin embargo, las técnicas descritas en esta divulgación pueden ser aplicables a la codificación de vídeo en general, y se pueden aplicar a aplicaciones inalámbricas y/o cableadas.

[0022] El codificador de vídeo 20 puede codificar el vídeo capturado, capturado previamente o generado por ordenador. Los datos de vídeo codificados se pueden transmitir directamente al dispositivo de destino 14 por medio de la interfaz de salida 22 del dispositivo de origen 12. Los datos de vídeo codificados se pueden almacenar también (o de forma alternativa) en el dispositivo de almacenamiento 34 para un posterior acceso por parte del dispositivo de destino 14 u otros dispositivos, para su decodificación y/o reproducción.

[0023] El dispositivo de destino 14 incluye una interfaz de entrada 28, un decodificador de vídeo 30 y un dispositivo de visualización 32. Como se explicará con mayor detalle a continuación, el decodificador de vídeo 30 puede configurarse para realizar las técnicas de ARP descritas en esta divulgación. En algunos casos, la interfaz de entrada 28 puede incluir un receptor y/o un módem. La interfaz de entrada 28 del dispositivo de destino 14 recibe los datos de vídeo codificados por el enlace 16. Los datos de vídeo codificados, comunicados por el enlace 16, o proporcionados en el dispositivo de almacenamiento 34, pueden incluir una diversidad de elementos sintácticos generados por el codificador de vídeo 20, para su uso por un decodificador de vídeo, tal como el decodificador de vídeo 30, en la decodificación de los datos de vídeo. Dichos elementos sintácticos se pueden incluir con los datos de vídeo codificados transmitidos en un medio de comunicación, almacenados en un medio de almacenamiento o almacenados en un servidor de ficheros.

[0024] El dispositivo de visualización 32 puede estar integrado en, o ser externo a, el dispositivo de destino 14. En algunos ejemplos, el dispositivo de destino 14 puede incluir un dispositivo de visualización integrado y también estar configurado para interconectarse con un dispositivo de visualización externo. En otros ejemplos, el dispositivo de destino 14 puede ser un dispositivo de visualización. En general, el dispositivo de visualización 32 exhibe los datos de vídeo decodificados ante un usuario y puede comprender cualquiera entre una variedad de dispositivos de visualización, tales como una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de visualización.

[0025] Aunque no se muestra en la figura 1, en algunos aspectos, tanto el codificador de vídeo 20 como el decodificador de vídeo 30 pueden estar integrados en un codificador y decodificador de audio, y pueden incluir unidades MUX-DEMUX adecuadas, u otro tipo de hardware y software, para gestionar la codificación tanto de audio como de vídeo en un flujo de datos común o en flujos de datos individuales. Si procede, en algunos ejemplos, las unidades MUX-DEMUX pueden ser conformes al protocolo multiplexador ITU H.223 o a otros protocolos, tales como el protocolo de datagramas de usuario (UDP).

[0026] El codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementarse, cada uno, como cualquiera entre una variedad de circuitos codificadores adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), formaciones de compuertas programables en el terreno (FPGA), lógica discreta, software, hardware, firmware o combinaciones cualesquiera de los mismos. Por ejemplo, las técnicas descritas en esta divulgación se pueden describir desde la perspectiva de un aparato o un dispositivo. Como ejemplo, el aparato o dispositivo puede incluir el decodificador de vídeo 30 (por ejemplo, el dispositivo de destino 14 como parte de un dispositivo de comunicación inalámbrica), y el decodificador de vídeo 30 puede incluir uno o más procesadores configurados para implementar las técnicas descritas en esta divulgación (por ejemplo, decodificar datos de vídeo de acuerdo a las técnicas descritas en esta divulgación). Como otro ejemplo, el aparato o dispositivo puede incluir un microprocesador o un circuito integrado (IC) que incluye el decodificador de vídeo 30, y el microprocesador o IC puede ser parte del dispositivo de destino 14 u otro tipo de dispositivo. Lo mismo se puede aplicar al codificador de vídeo 20 (es decir, un aparato o dispositivo como el dispositivo de origen 12 y/o un microcontrolador o IC incluye el codificador de vídeo 20, donde el codificador de vídeo 20 se configura para codificar datos de vídeo de acuerdo a las técnicas descritas en esta divulgación).

[0027] 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 utilizando uno o más procesadores para realizar las técnicas de esta divulgación. Cada uno entre el codificador de vídeo 20 y el decodificador de vídeo 30 se puede incluir en uno o más codificadores o decodificadores, cualquiera de los cuales se puede integrar como parte de un codificador/decodificador (CÓDEC) combinado en un dispositivo respectivo.

[0028] Una secuencia de vídeo normalmente incluye una serie de imágenes de vídeo desde una vista. Un grupo de imágenes (GOP) comprende en general una serie de una o más imágenes de vídeo. Un GOP puede incluir datos sintácticos en una cabecera del GOP, una cabecera de una o más imágenes del GOP o en otras ubicaciones, que describen una serie de imágenes incluidas en el GOP. Cada imagen puede incluir datos sintácticos de imagen que describen una modalidad de codificación para la imagen respectiva. Un codificador de vídeo 20 opera habitualmente sobre bloques de vídeo dentro de imágenes de vídeo individuales a fin de codificar los datos de vídeo. Un bloque de vídeo puede corresponder a un macrobloque, una partición de un macrobloque y, posiblemente, un subbloques de

una partición, como se define en la norma H.264. Los bloques de vídeo pueden tener tamaños fijos o variables y pueden diferir en tamaño de acuerdo a una norma de codificación especificada. Cada imagen de vídeo puede incluir una pluralidad de fragmentos. Cada fragmento puede incluir una pluralidad de bloques.

5 **[0029]** Como ejemplo, la norma UIT-T H.264 presta soporte a la intrapredicción en diversos tamaños de bloque, tales como 16 por 16, 8 por 8 o 4 por 4 para componentes de luma, y 8x8 para componentes de croma, así como la interpredicción en diversos tamaños de bloque, tales como 16x16, 16x8, 8x16, 8x8, 8x4, 4x8 y 4x4 para componentes de luma y tamaños correspondientes ajustados a escala para componentes de croma. En esta divulgación, "NxN" y "N por N" se pueden usar de manera intercambiable para referirse a las dimensiones de píxel del bloque en lo que
10 respecta a las dimensiones vertical y horizontal, (por ejemplo, 16x16 píxeles o 16 por 16 píxeles). En general, un bloque de 16x16 tendrá 16 píxeles en una dirección vertical ($y = 16$) y 16 píxeles en una dirección horizontal ($x = 16$). Asimismo, un bloque de NxN tiene, en general, N píxeles en una dirección vertical y N píxeles en una dirección horizontal, donde N representa un valor entero no negativo. Los píxeles de un bloque se pueden disponer en filas y columnas. Además, no es necesario que los bloques tengan necesariamente el mismo número de píxeles en la
15 dirección horizontal y en la dirección vertical. Por ejemplo, los bloques pueden comprender NxM píxeles, donde M no es necesariamente igual a N.

[0030] Cuando el bloque se codifica en intramodalidad (por ejemplo, intrapredicción), el bloque puede incluir datos que describen una modalidad de intrapredicción para el bloque. Como otro ejemplo, cuando el bloque se codifica en intermodalidad (por ejemplo, interpredicción), el bloque puede incluir información que define un vector de movimiento para el bloque. Este vector de movimiento se refiere a una imagen de referencia en la misma vista (por ejemplo, un vector de movimiento temporal), o se refiere a una imagen de referencia en otra vista (por ejemplo, un vector de movimiento de disparidad). Los datos que definen el vector de movimiento para un bloque describen, por ejemplo, una
20 componente horizontal del vector de movimiento, una componente vertical del vector de movimiento, una resolución para el vector de movimiento (por ejemplo, una precisión de un cuarto de píxel o una precisión de un octavo de píxel). Además, cuando se interpredice, el bloque puede incluir información de índice de referencia, tal como una imagen de referencia a la que apunta el vector de movimiento, y/o una lista de imágenes de referencia (por ejemplo, ListalngRef0 o ListalngRef1) para el vector de movimiento.

30 **[0031]** En la norma H.264, tras la codificación intrapredicativa o interpredicativa, el codificador de vídeo 20 calcula los datos residuales para los macrobloques. Los datos residuales pueden corresponder a diferencias de píxeles entre píxeles de la imagen no codificada y valores de predicción para el macrobloque en la norma H.264.

[0032] Tras cualquier transformación para producir coeficientes de transformación, el codificador de vídeo 20 realiza la cuantización de los coeficientes de transformación, en algunos ejemplos. La cuantización se refiere, en general, a un proceso en el que los coeficientes de transformación se cuantifican para reducir posiblemente la cantidad de datos usados para representar los coeficientes, proporcionando compresión adicional. El proceso de cuantización reduce la profundidad de bits asociada a algunos o a la totalidad de los coeficientes. Por ejemplo, un valor de n bits se redondea a la baja hasta un valor de m bits durante la cuantización, donde n es mayor que m .
35

[0033] En algunos ejemplos, el codificador de vídeo 20 utiliza un orden de recorrido predefinido para explorar los coeficientes de transformación cuantizados, para producir un vector en serie que se pueda someter a codificación por entropía. En otros ejemplos, el codificador de vídeo 20 realiza un recorrido adaptativo. Después de recorrer los coeficientes de transformación cuantizados para formar un vector unidimensional, en algunos ejemplos, el codificador de vídeo 20 codifica por entropía el vector unidimensional de acuerdo a la codificación de longitud variable adaptativa al contexto (CAVLC), la codificación aritmética binaria adaptativa al contexto (CABAC), la codificación aritmética binaria adaptativa al contexto basada en la sintaxis (SBAC), la codificación por entropía por división de intervalos de probabilidad (PIPE) u otra metodología de codificación por entropía, como unos pocos ejemplos. El codificador de vídeo 20 también codifica por entropía elementos sintácticos asociados a los datos de vídeo codificados, para su uso
40 por el decodificador de vídeo 30 en la decodificación de los datos de vídeo.

[0034] Para realizar la CABAC, el codificador de vídeo 20 puede asignar un contexto dentro de un modelo de contexto a un símbolo a transmitir. El contexto se puede referir, por ejemplo, a si los valores contiguos del símbolo son distintos de cero o no. Para realizar la CAVLC, el codificador de vídeo 20 puede seleccionar un código de longitud variable para un símbolo a transmitir. Las palabras de código en la VLC se pueden construir de modo que los códigos relativamente más cortos correspondan a símbolos más probables, mientras que los códigos más largos correspondan a símbolos menos probables. De esta forma, el uso de la VLC puede lograr un ahorro en bits con respecto, por ejemplo, al uso de palabras de código de igual longitud para cada símbolo a transmitir. La determinación de la probabilidad se puede basar en un contexto asignado al símbolo.
55

[0035] El decodificador de vídeo 30 implementa la inversa de las técnicas del codificador de vídeo 20. Por ejemplo, el decodificador de vídeo 30 decodifica el flujo de bits de vídeo codificado y determina los bloques residuales mediante cuantización inversa y transformación inversa. El decodificador de vídeo 30 suma los bloques residuales con bloques de imágenes decodificadas previamente para determinar los valores de píxeles para los bloques dentro de la imagen.
60

65

- 5 **[0036]** Ciertas técnicas descritas en esta divulgación pueden ser realizadas tanto por el codificador de vídeo 20 como por el decodificador de vídeo 30. Como ejemplo, el codificador de vídeo 20 puede realizar la ARP como parte de la determinación de cómo codificar un bloque de datos de vídeo y / o puede realizar la ARP como parte de un bucle de decodificación en el codificador de vídeo. El decodificador de vídeo 30 puede realizar las mismas técnicas de ARP realizadas por el codificador de vídeo 20 como parte de la decodificación del bloque de vídeo. Esta divulgación puede referirse a veces al decodificador de vídeo 30 que realiza ciertas técnicas de ARP descritas en esta divulgación. Sin embargo, debería entenderse que, a menos que se indique lo contrario, dichas técnicas también pueden ser realizadas por el codificador de vídeo 20.
- 10 **[0037]** Como se ha descrito anteriormente, las técnicas descritas en esta divulgación están dirigidas a la codificación de vídeo tridimensional. Para comprender mejor las técnicas, a continuación se describen algunas técnicas de codificación de la H.264/AVC, la codificación de vídeo de vista múltiple desde la perspectiva de la ampliación H.264/MVC y la norma de Codificación de Vídeo de Alta Eficacia (HEVC) y técnicas de la 3D-AVC.
- 15 **[0038]** Para la norma H.264/Codificación Avanzada de Vídeo (AVC), la codificación o decodificación de vídeo (por ejemplo, la codificación) se implementa en macrobloques, donde un macrobloque representa una parte de una trama que se interpredice o se intrapredice (es decir, se codifica o decodifica por interpredicción o se codifica o decodifica por intrapredicción). Por ejemplo, en la norma H.264/AVC, cada inter-macrobloque (MB) (por ejemplo, macrobloque interpredicho) se puede dividir en cuatro formas diferentes: una partición de 16x16 MB, dos particiones de 16x8 MB, dos particiones de 8x16 MB o cuatro particiones de 8x8 MB. Las diferentes particiones de MB en un MB pueden tener diferentes valores de índice de referencia para cada dirección (es decir, ListalMgRef0 o ListalMgRef1). Cuando un MB no se divide en múltiples (más de 1) particiones de MB, tiene solamente un vector de movimiento para toda la partición de MB en cada dirección.
- 20 **[0039]** Como parte de la codificación de vídeo (codificación o decodificación), el codificador de vídeo 20/30 puede configurarse para construir una o dos listas de imágenes de referencia, denominadas ListalMgRef0 y ListalMgRef1. La(s) lista(s) de imágenes de referencia identifica(n) imágenes de referencia que se pueden usar para interpredicar macrobloques de una trama o un fragmento. Por ejemplo, el codificador de vídeo 20 puede señalar un índice de referencia y un identificador de lista de imágenes de referencia. El decodificador de vídeo 30 puede recibir el índice de referencia y el identificador de la lista de imágenes de referencia y determinar la imagen de referencia que se va a usar para decodificar por interpredicción el macrobloque actual a partir del índice de referencia y del identificador de la lista de imágenes de referencia.
- 25 **[0040]** Cuando un MB se divide en cuatro particiones de 8x8 MB, cada partición de 8x8 MB se puede dividir además en subbloques. Existen cuatro formas diferentes de obtener subbloques a partir de una partición de 8x8 MB: un sub-bloque de 8x8, dos subbloques de 8x4, dos subbloques de 4x8 o cuatro subbloques de 4x4. Cada sub-bloque puede tener un vector de movimiento diferente en cada dirección, pero comparte el mismo índice de imagen de referencia para cada dirección. La manera en que una partición de 8x8 MB se divide en subbloques se denomina partición de subbloques.
- 30 **[0041]** Esta divulgación generalmente utilizará el término bloque para referirse a cualquier bloque de datos de vídeo. Por ejemplo, en el contexto de la codificación según la norma H.264 y sus extensiones, un bloque puede referirse a cualquiera entre macrobloques, particiones de macrobloques, subbloques u otros tipos cualesquiera de bloques. En el contexto de la HEVC y sus extensiones, un bloque puede referirse a cualquiera de las PU, TU, CU u otros tipos cualesquiera de bloques. Un sub-bloque, como se usa en esta divulgación, generalmente se refiere a cualquier parte de un bloque más grande. Un sub-bloque mismo también puede ser mencionado simplemente como un bloque.
- 35 **[0042]** Para la codificación de vídeo de vista múltiple, existen múltiples normas diferentes de codificación de vídeo. Para evitar confusiones, cuando esta divulgación describe genéricamente la codificación de vídeo de vista múltiple, esta divulgación usa la frase "codificación de vídeo de vista múltiple". En general, en la codificación de vídeo de vista múltiple, hay una vista base y una o más vistas no base o dependientes. La vista base es totalmente decodificable sin referencia a ninguna de las vistas dependientes (es decir, la vista base solamente se inter-predice con vectores de movimiento temporal). Esto permite que un códec que no está configurado para la codificación de vídeo de vista múltiple reciba todavía al menos una vista que sea totalmente decodificable (es decir, la vista base se puede extraer y las otras vistas se pueden descartar, permitiendo que un decodificador no configurado para la codificación de vídeo de vista múltiple decodifique, sin embargo, el contenido del vídeo, aunque sea sin experiencia tridimensional). Las una o más vistas dependientes se pueden interpredicar con respecto a la vista base o con respecto a otra vista dependiente (es decir, predecir con compensación de disparidad), o con respecto a otras imágenes en la misma vista (es decir, predecir con compensación de movimiento).
- 40 **[0043]** Si bien la "codificación de vídeo de vista múltiple" se usa genéricamente, el acrónimo MVC está asociado a una ampliación de la H.264/AVC. En consecuencia, cuando la divulgación usa el acrónimo MVC, la divulgación se refiere específicamente a la ampliación de la norma de codificación de vídeo H.264/AVC. La ampliación MVC de la H.264/AVC se basa en los vectores de movimiento de disparidad como otro tipo de vector de movimiento además de los vectores de movimiento temporal. Otra norma de codificación de vídeo, denominada MVC más profundidad (MVC+D), también ha sido desarrollada por JCT-3V y MPEG. MVC+D aplica las mismas herramientas de codificación
- 45
50
55
60
65

de bajo nivel que las de MVC, tanto para la textura como para la profundidad, siendo la decodificación de la profundidad independiente de la decodificación de la textura, y viceversa. Por ejemplo, en la MVC, una trama está representada solamente por un componente de vista, denominado componente de vista de textura o, simplemente, textura. En MVC+D, existen dos componentes de vista: el componente de vista de textura y el componente de vista de profundidad o, simplemente, textura y profundidad. Por ejemplo, en MVC+D, cada vista incluye una vista de textura y una vista de profundidad, donde la vista incluye una pluralidad de componentes de vista, la vista de textura incluye una pluralidad de componentes de vista de textura y la vista de profundidad incluye una pluralidad de componentes de vista de profundidad.

[0044] Cada componente de vista de textura está asociado a un componente de vista de profundidad para formar un componente de vista de una vista. El componente de vista de profundidad representa la profundidad relativa de los objetos en el componente de vista de textura. En MVC+D, el componente de vista de profundidad y el componente de vista de textura son decodificables por separado. Por ejemplo, el decodificador de vídeo 30 puede implementar dos casos de un códec de MVC, en los que un primer códec decodifica los componentes de vista de textura y un segundo códec decodifica los componentes de vista de profundidad. Estos dos códecs se pueden ejecutar de forma independiente uno del otro, porque los componentes de vista de textura y los componentes de vista de profundidad se codifican por separado.

[0045] En MVC+D, un componente de vista de profundidad siempre sigue inmediatamente al componente de vista de textura asociado (por ejemplo, correspondiente). De esta manera, la MVC+D presta soporte a la codificación de textura en primer lugar, donde el componente de vista de textura se decodifica antes del componente de vista de profundidad.

[0046] Un componente de vista de textura y su componente de vista de profundidad asociado (por ejemplo, correspondiente) pueden incluir el mismo valor de recuento de orden de imagen (POC) y el mismo id_vista (es decir, el valor de POC y el id_vista de un componente de vista de textura y su componente asociado de vista de profundidad son el mismo). El valor de POC indica el orden de visualización del componente de vista de textura y el id_vista indica la vista a la que pertenecen el componente de vista de textura y el componente de vista de profundidad.

[0047] La figura 2 muestra un típico orden de decodificación de la MVC (es decir, orden del flujo de bits). La disposición del orden de decodificación se denomina codificación de tiempo en primer lugar. Se debe tener en cuenta que el orden de decodificación de las unidades de acceso puede no ser idéntico al orden de salida o de visualización. En la figura 2, S0 a S7 se refieren, cada uno, a diferentes vistas del vídeo de vista múltiple. Cada una entre T0 a T8 representa una instancia de tiempo de salida. Una unidad de acceso puede incluir las imágenes codificadas de todas las 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.

[0048] En aras de la brevedad, la divulgación puede utilizar 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 puede incluir 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*.

[0049] Como se ha expuesto anteriormente, en el contexto de esta divulgación, el componente de vista, el componente de vista de textura y el componente de vista de profundidad pueden denominarse, en general, una capa. En la figura 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. Cada conjunto incluye dos imágenes: una imagen se denomina componente de vista de textura y la otra imagen se denomina componente de vista de profundidad. El componente de vista de textura y el componente de vista de profundidad dentro de un conjunto de imágenes de una vista se pueden considerar como correspondientes entre sí. Por ejemplo, el componente de vista de textura dentro de un conjunto de imágenes de una vista se considera correspondiente al componente de vista de profundidad dentro del conjunto de las imágenes de la vista, y viceversa (es decir, el componente de vista de profundidad corresponde a su componente de vista de textura en el conjunto, y viceversa). 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.

[0050] El componente de vista de textura incluye el contenido de imagen real que se muestra. Por ejemplo, el componente de vista de textura puede incluir los componentes de luma (Y) y croma (Cb y Cr). El componente de vista

de profundidad puede indicar profundidades relativas de los píxeles en su correspondiente componente de vista de textura. Como analogía ejemplar, el componente de vista de profundidad es como una imagen en escala de grises que incluye solo 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.

[0051] Por ejemplo, un píxel blanco puro en el componente de vista de profundidad indica que su correspondiente píxel, o correspondientes píxeles, en el correspondiente componente de vista de textura, está(n) más cerca, desde la perspectiva del observador, y un píxel negro puro en el componente de vista de profundidad indica que su correspondiente píxel, o correspondientes píxeles, en el correspondiente componente de vista de textura, está(n) más lejos, desde la perspectiva del observador. Los diversos tonos de gris entre negro y blanco indican diferentes niveles de profundidad. Por ejemplo, un píxel muy gris en el componente de vista de profundidad indica que su píxel correspondiente en el componente de vista de textura está más alejado que un píxel ligeramente gris en el componente de vista de profundidad. Dado que solamente es necesaria la escala de grises para identificar la profundidad de los píxeles, el componente de vista de profundidad no necesita incluir componentes de croma, ya que los valores de color para el componente de vista de profundidad pueden no tener ningún sentido. La explicación anterior pretende ser una analogía con fines de relacionar imágenes de profundidad con imágenes de textura. Los valores de profundidad en una imagen de profundidad, de hecho, no representan matices de gris pero, de hecho, representan valores de profundidad de 8 bits, o de otro tamaño en bits.

[0052] La componente de visualización de profundidad que utiliza solo valores de luma (por ejemplo, valores de intensidad) para identificar la profundidad se proporciona con fines ilustrativos y no debería considerarse limitativa. En otros ejemplos, se puede utilizar cualquier técnica para indicar las profundidades relativas de los píxeles en el componente de vista de textura.

[0053] La figura 3 muestra una típica estructura de predicción de MVC (que incluye tanto la predicción entre imágenes dentro de cada vista como la predicción entre vistas) para la codificación de vídeo de vista múltiple. Las direcciones de predicción se indican mediante flechas, utilizando, el objeto apuntado, el objeto desde el que se apunta como referencia de predicción. En la MVC, la predicción entre vistas tiene soporte de la compensación de movimiento de disparidad, que usa la sintaxis de la compensación de movimiento de la H.264/AVC, pero permite que una imagen en una vista diferente sea usada como imagen de referencia.

[0054] En el ejemplo de la figura 3, se ilustran ocho vistas (que tienen identificadores de vista "S0" a "S7") y se ilustran doce ubicaciones temporales ("T0" a "T11") para cada vista. Es decir, cada fila en la figura 3 corresponde a una vista, mientras que cada columna indica una ubicación temporal.

[0055] Aunque la MVC tiene una denominada vista base, que es decodificable mediante los decodificadores de la H.264/AVC, y los pares de vistas estéreo también pueden tener soporte por parte de la MVC, la ventaja de la MVC es que podría prestar soporte a un ejemplo que usa más de dos vistas como una entrada de vídeo tridimensional y decodifica este vídeo tridimensional representado por las múltiples vistas. Un representador de un cliente que tiene un decodificador de MVC puede esperar contenido de vídeo tridimensional con múltiples vistas.

[0056] Las imágenes en la figura 3 se indican en la intersección de cada fila y cada columna. La norma H.264/AVC puede usar el término trama para representar una parte del vídeo. Esta divulgación puede usar los términos imagen y trama de forma intercambiable.

[0057] Las imágenes en la figura 3 se ilustran usando un bloque que incluye una letra, designando la letra si la imagen correspondiente está intracodificada (es decir, es una imagen I), o intercodificada en una dirección (es decir, como una imagen P) o en múltiples direcciones (es decir, como una imagen B). En general, las predicciones se indican mediante flechas, donde las imágenes a las que se apunta usan la imagen desde la que se apunta como referencia de predicción. Por ejemplo, la imagen P de la vista S2 en la ubicación temporal T0 se predice a partir de la imagen I de la vista S0 en la ubicación temporal T0.

[0058] Al igual que con la codificación de vídeo de vista única, las imágenes de una secuencia de vídeo de codificación de vídeo de múltiples vistas pueden codificarse predictivamente con respecto a imágenes en diferentes ubicaciones temporales. Por ejemplo, la imagen b de la vista S0 en la ubicación temporal T1 tiene una flecha apuntando a la misma desde la imagen I de la vista S0 en la ubicación temporal T0, indicando que la imagen b se predice a partir de la imagen I. Adicionalmente, sin embargo, en el contexto de la codificación de vídeo de múltiples vistas, las imágenes se pueden predecir entre vistas. Es decir, un componente de vista puede usar los componentes de vista en otras vistas como referencia. En la MVC, por ejemplo, la predicción entre vistas se realiza como si el componente de vista en otra vista es una referencia de interpredicción. Las referencias potenciales entre vistas se señalizan en la ampliación 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 de predicción entre vistas. La predicción entre vistas también es una característica de la ampliación propuesta de vista múltiple de la HEVC, incluyendo la 3D-HEVC (vista múltiple más profundidad).

- 5 **[0059]** La figura 3 proporciona varios ejemplos de predicción entre vistas. Las imágenes de la vista S1, en el ejemplo de la figura 3, se ilustran como predichas a partir de imágenes en diferentes ubicaciones temporales de la vista S1, así como predichas entre vistas a partir de imágenes de las vistas S0 y S2 en las mismas ubicaciones temporales. Por ejemplo, la imagen b de la vista S1 en la ubicación temporal T1 se predice a partir de cada una de las imágenes B de la vista S1 en las ubicaciones temporales T0 y T2, así como las imágenes b de las vistas S0 y S2 en la ubicación temporal T1.
- 10 **[0060]** En algunos ejemplos, la figura 3 puede verse como una ilustración de los componentes de la vista de textura. Por ejemplo, las imágenes l, P, B y b, ilustradas en la figura 2, se pueden considerar componentes de vista de textura para cada una de las vistas. De acuerdo a las técnicas descritas en esta divulgación, para cada uno de los componentes de vista de textura ilustrados en la figura 3 existe un correspondiente componente de vista de profundidad. En algunos ejemplos, los componentes de vista de profundidad se pueden predecir de una manera similar a la ilustrada en la figura 3 para los correspondientes componentes de vista de textura.
- 15 **[0061]** La codificación de dos vistas también puede tener soporte de la MVC. Una de las ventajas de la MVC es que un codificador de MVC puede tomar más de dos vistas como una entrada de vídeo tridimensional y un decodificador de MVC puede decodificar dicha representación de vista múltiple. De tal modo, cualquier representador con un decodificador de MVC puede decodificar contenidos de vídeo tridimensionales con más de dos vistas.
- 20 **[0062]** Como se ha expuesto anteriormente, en la MVC, se permite la predicción entre vistas entre imágenes en la misma unidad de acceso (lo que quiere decir, en algunos casos, con la misma instancia de tiempo). Cuando se codifica una imagen en una de las vistas no básicas, se puede añadir una imagen a una lista de imágenes de referencia, si está en una vista diferente pero dentro de una misma instancia de tiempo. Una imagen de referencia de predicción entre vistas se puede poner en cualquier posición de una lista de imágenes de referencia, tal como cualquier imagen de referencia de interpredicción. Como se muestra en la figura 3, un componente de vista puede usar los componentes de vista en otras vistas como referencia. En la MVC, la predicción entre vistas se realiza como si el componente de vista en otra vista fuera una referencia de interpredicción.
- 25 **[0063]** En la MVC, se permite la predicción entre vistas entre imágenes en la misma unidad de acceso (es decir, con la misma instancia de tiempo). Cuando se codifica una imagen en una de las vistas no básicas, se puede añadir una imagen a una lista de imágenes de referencia si está en una vista diferente pero con una misma instancia de tiempo. Una imagen de referencia de predicción entre vistas se puede poner en cualquier posición de una lista de imágenes de referencia, tal como cualquier imagen de referencia de interpredicción.
- 30 **[0064]** Como se muestra en la figura 3, un componente de vista puede usar los componentes de vista en otras vistas como referencia. Esto se llama predicción entre vistas. En la MVC, la predicción entre vistas se realiza como si el componente de vista en otra vista fuera una referencia de interpredicción.
- 35 **[0065]** En el contexto de la codificación de vídeo de vista múltiple, hay dos tipos de vectores de movimiento: uno es un vector de movimiento normal que apunta a imágenes de referencia temporales. La interpredicción temporal correspondiente es la predicción compensada por movimiento (MCP). El otro tipo de vector de movimiento es un vector de movimiento de disparidad que apunta a imágenes en una vista diferente (es decir, imágenes de referencia entre vistas). La interpredicción correspondiente es la predicción compensada por disparidad (DCP).
- 40 **[0066]** El decodificador de vídeo 30 puede decodificar vídeo usando múltiples modalidades de intercodificación de la HEVC. En la norma HEVC, hay dos modalidades de interpredicción, llamadas, respectivamente, modalidades de fusión (la omisión se considera un caso especial de fusión) y de predicción avanzada de vectores de movimiento (AMVP), para una unidad de predicción (PU). En la modalidad de AMVP o de fusión, el decodificador de vídeo 30 mantiene una lista de candidatos de vectores de movimiento (MV) para múltiples predictores de vectores de movimiento. El (los) vector(es) de movimiento, así como los índices de referencia en la modalidad de fusión, de la PU actual pueden ser generados tomando un candidato de la lista de candidatos de MV.
- 45 **[0067]** La lista de candidatos de MV contiene, por ejemplo, hasta cinco candidatos para la modalidad de fusión y solo dos candidatos para la modalidad de AMVP. Un candidato de fusión puede contener un conjunto de información de movimiento, por ejemplo, vectores de movimiento correspondientes a ambas listas de imágenes de referencia (lista 0 y lista 1) y los índices de referencia. Si un candidato de fusión se identifica mediante un índice de fusión, las imágenes de referencia se utilizan para la predicción de los bloques actuales, como así también se determinan los vectores de movimiento asociados. Sin embargo, en la modalidad de AMVP, para cada dirección de predicción potencial de la lista 0 o la lista 1, un índice de referencia necesita ser señalado explícitamente, junto con un índice de MVP para la lista de candidatos de MV, ya que el candidato de AMVP solo contiene un vector de movimiento. En la modalidad de AMVP, la diferencia de vectores de movimiento, entre el vector de movimiento seleccionado y el predictor de vector de movimiento correspondiente al índice de MVP, se señala adicionalmente. Como se puede ver en lo que antecede, un candidato de fusión corresponde a un conjunto completo de información de movimiento, mientras que un candidato de AMVP contiene solo un vector de movimiento para una dirección de predicción e índice de referencia específicos.
- 50 **[0067]** La lista de candidatos de MV contiene, por ejemplo, hasta cinco candidatos para la modalidad de fusión y solo dos candidatos para la modalidad de AMVP. Un candidato de fusión puede contener un conjunto de información de movimiento, por ejemplo, vectores de movimiento correspondientes a ambas listas de imágenes de referencia (lista 0 y lista 1) y los índices de referencia. Si un candidato de fusión se identifica mediante un índice de fusión, las imágenes de referencia se utilizan para la predicción de los bloques actuales, como así también se determinan los vectores de movimiento asociados. Sin embargo, en la modalidad de AMVP, para cada dirección de predicción potencial de la lista 0 o la lista 1, un índice de referencia necesita ser señalado explícitamente, junto con un índice de MVP para la lista de candidatos de MV, ya que el candidato de AMVP solo contiene un vector de movimiento. En la modalidad de AMVP, la diferencia de vectores de movimiento, entre el vector de movimiento seleccionado y el predictor de vector de movimiento correspondiente al índice de MVP, se señala adicionalmente. Como se puede ver en lo que antecede, un candidato de fusión corresponde a un conjunto completo de información de movimiento, mientras que un candidato de AMVP contiene solo un vector de movimiento para una dirección de predicción e índice de referencia específicos.
- 55 **[0067]** La lista de candidatos de MV contiene, por ejemplo, hasta cinco candidatos para la modalidad de fusión y solo dos candidatos para la modalidad de AMVP. Un candidato de fusión puede contener un conjunto de información de movimiento, por ejemplo, vectores de movimiento correspondientes a ambas listas de imágenes de referencia (lista 0 y lista 1) y los índices de referencia. Si un candidato de fusión se identifica mediante un índice de fusión, las imágenes de referencia se utilizan para la predicción de los bloques actuales, como así también se determinan los vectores de movimiento asociados. Sin embargo, en la modalidad de AMVP, para cada dirección de predicción potencial de la lista 0 o la lista 1, un índice de referencia necesita ser señalado explícitamente, junto con un índice de MVP para la lista de candidatos de MV, ya que el candidato de AMVP solo contiene un vector de movimiento. En la modalidad de AMVP, la diferencia de vectores de movimiento, entre el vector de movimiento seleccionado y el predictor de vector de movimiento correspondiente al índice de MVP, se señala adicionalmente. Como se puede ver en lo que antecede, un candidato de fusión corresponde a un conjunto completo de información de movimiento, mientras que un candidato de AMVP contiene solo un vector de movimiento para una dirección de predicción e índice de referencia específicos.
- 60 **[0067]** La lista de candidatos de MV contiene, por ejemplo, hasta cinco candidatos para la modalidad de fusión y solo dos candidatos para la modalidad de AMVP. Un candidato de fusión puede contener un conjunto de información de movimiento, por ejemplo, vectores de movimiento correspondientes a ambas listas de imágenes de referencia (lista 0 y lista 1) y los índices de referencia. Si un candidato de fusión se identifica mediante un índice de fusión, las imágenes de referencia se utilizan para la predicción de los bloques actuales, como así también se determinan los vectores de movimiento asociados. Sin embargo, en la modalidad de AMVP, para cada dirección de predicción potencial de la lista 0 o la lista 1, un índice de referencia necesita ser señalado explícitamente, junto con un índice de MVP para la lista de candidatos de MV, ya que el candidato de AMVP solo contiene un vector de movimiento. En la modalidad de AMVP, la diferencia de vectores de movimiento, entre el vector de movimiento seleccionado y el predictor de vector de movimiento correspondiente al índice de MVP, se señala adicionalmente. Como se puede ver en lo que antecede, un candidato de fusión corresponde a un conjunto completo de información de movimiento, mientras que un candidato de AMVP contiene solo un vector de movimiento para una dirección de predicción e índice de referencia específicos.
- 65 **[0067]** La lista de candidatos de MV contiene, por ejemplo, hasta cinco candidatos para la modalidad de fusión y solo dos candidatos para la modalidad de AMVP. Un candidato de fusión puede contener un conjunto de información de movimiento, por ejemplo, vectores de movimiento correspondientes a ambas listas de imágenes de referencia (lista 0 y lista 1) y los índices de referencia. Si un candidato de fusión se identifica mediante un índice de fusión, las imágenes de referencia se utilizan para la predicción de los bloques actuales, como así también se determinan los vectores de movimiento asociados. Sin embargo, en la modalidad de AMVP, para cada dirección de predicción potencial de la lista 0 o la lista 1, un índice de referencia necesita ser señalado explícitamente, junto con un índice de MVP para la lista de candidatos de MV, ya que el candidato de AMVP solo contiene un vector de movimiento. En la modalidad de AMVP, la diferencia de vectores de movimiento, entre el vector de movimiento seleccionado y el predictor de vector de movimiento correspondiente al índice de MVP, se señala adicionalmente. Como se puede ver en lo que antecede, un candidato de fusión corresponde a un conjunto completo de información de movimiento, mientras que un candidato de AMVP contiene solo un vector de movimiento para una dirección de predicción e índice de referencia específicos.

[0068] Como se ha presentado anteriormente, el decodificador de vídeo 30 puede decodificar vídeo codificado de acuerdo a la norma de codificación de vídeo tridimensional basada en la HEVC. Actualmente, un Equipo de Colaboración Conjunta en Codificación de vídeo tridimensional (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 de múltiples vistas basado en la HEVC (MV-HEVC) y otra parte, la codificación de vídeo tridimensional basada en la HEVC (3D-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. El software 3D-HTM más reciente para la 3D-HEVC se puede descargar desde el siguiente enlace:

[3D-HTM versión 9.0r1]: https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-9.0r1/

[0069] La más reciente descripción del software de referencia ha de estar disponible de la siguiente manera: artículo de Li Zhang, Gerhard Tech, Krzysztof Wegner y Sehoon Yea, "Test Model 6 of 3D-HEVC and MV-HEVC" ["Modelo de prueba 6 de la 3D-HEVC y la MV-HEVC"], JCT3V-F1005, Equipo de colaboración conjunta en el desarrollo de la extensión de codificación de vídeo tridimensional del UIT-T SG 16 WP 3 e ISO / IEC JTC 1 / SC 29 / WG 11, 6ª conferencia: Ginebra, CH, noviembre de 2013. Podría descargarse desde el siguiente enlace: http://phenix.it-sudparis.eu/jct2/doc_end_user/current_document.php?id=1636

[0070] El más reciente borrador de trabajo de la 3D-HEVC está disponible de la siguiente manera: artículo de Gerhard Tech, Krzysztof Wegner, Ying Chen y Sehoon Yea, "3D-HEVC Test Model 2" ["Modelo de prueba 2 de la 3D-HEVC"], JCT3V-F1001, equipo conjunto de colaboración en desarrollo de ampliación de codificación de vídeo tridimensional de ITU-T SG 16 WP 3 e ISO / IEC JTC 1 / SC 29 / WG 11, 6ª Conferencia. Ginebra, CH, noviembre de 2013. Se podría descargar desde el siguiente enlace: http://phenix.it-sudparis.eu/jct2/doc_end_user/documents/6_Geneva/wg11/JCT3V-F1001-v4.zip

[0071] Como parte de la decodificación de vídeo de acuerdo a la norma HEVC, el decodificador de vídeo 30 puede configurarse para realizar la Obtención de vectores de disparidad basándose en bloques vecinos (NBDV). La NBDV es un procedimiento de obtención de vectores de disparidad en la 3D-HEVC que utiliza el orden de codificación de textura en primer lugar para todas las vistas. En el diseño actual de la 3D-HEVC, el vector de disparidad obtenido a partir de la NBDV se puede refinar aún más recuperando los datos de profundidad desde el mapa de profundidades de la vista de referencia.

[0072] El decodificador de vídeo 30 puede usar un vector de disparidad (DV) como estimador del desplazamiento entre dos vistas. Debido a que los bloques vecinos comparten casi la misma información de movimiento/disparidad en la codificación de vídeo, el bloque actual puede usar la información de vectores de movimiento de bloques vecinos como un buen predictor. Siguiendo esta idea, la NBDV utiliza la información de disparidad adyacente para estimar el vector de disparidad en diferentes vistas.

[0073] Como parte de la realización de la NBDV, primero se definen varios bloques vecinos, espaciales y temporales. El decodificador de vídeo 30 puede entonces verificar cada uno de ellos en un orden predefinido, determinado por la prioridad de la correlación entre el bloque actual y el bloque candidato. Una vez que se encuentra en los candidatos un vector de movimiento de disparidad (es decir, el vector de movimiento apunta a una imagen de referencia entre vistas), el vector de movimiento de disparidad se convierte en un vector de disparidad y también se devuelve el índice de orden de vista asociada. Se utilizan dos conjuntos de bloques contiguos. Un conjunto es de bloques de vecindad espacial y el otro conjunto es de bloques de vecindad temporal.

[0074] La 3D-HEVC adoptó por primera vez el procedimiento de NBDV propuesto en el documento JCT3V-A0097. Se incluyeron vectores de disparidad implícita con una NBDV simplificada en el documento JCTVC-A0126. Además, en el documento JCT3V-B0047, la NBDV se simplifica aún más al eliminar los vectores de disparidad implícitos almacenados en el almacén temporal de imágenes decodificadas, pero también se mejora una ganancia de codificación con la selección de imágenes de RAP. Los siguientes documentos describen aspectos de la 3D-HEVC y de la NBDV.

- JCT3V-A0097: 3D-CE5.h: Disparity vector generation results, L. Zhang, Y. Chen, M. Karczewicz (Qualcomm) [Resultados de la generación de vectores de disparidad, L. Zhang, Y. Chen, M. Karczewicz (Qualcomm)]
- JCT3V-A0126: 3D-CE5.h: Simplification of disparity vector derivation for HEVC-based 3D video coding, J. Sung, M. Koo, S. Yea (LG) [Simplificación de la obtención de vectores de disparidad para la codificación de vídeo tridimensional basándose en la HEVC, J. Sung, M. Koo, S. Yea (LG)]
- JCT3V-B0047: 3D-CE5.h related: Improvements for disparity vector derivation, J. Kang, Y. Chen, L. Zhang, M. Karczewicz (Qualcomm) [Mejoras para la obtención de vectores de disparidad, J. Kang, Y. Chen, L. Zhang, M. Karczewicz (Qualcomm)]
- JCT3V-D0181: CE2: CU-based Disparity Vector Derivation in 3D-HEVC, J. Kang, Y. Chen, L. Zhang, M. Karczewicz (Qualcomm) [Obtención de vectores de disparidad basándose en CU en la 3D-HEVC, J. Kang, Y. Chen, L. Zhang, M. Karczewicz (Qualcomm)]

- 5 **[0075]** La figura 4 muestra un ejemplo de vecinos vectoriales de movimiento espacial con respecto a una unidad de codificación. En algunas implementaciones de la NBDV, se utilizan cinco bloques de adyacencia espacial para la obtención de vectores de disparidad. Son los bloques abajo a la izquierda, a la izquierda, arriba a la derecha, arriba y arriba a la izquierda de la unidad de codificación (CU) que abarcan la unidad de predicción actual (PU), según lo indicado por A0, A1, B0, B1 o B2, como se muestra en la figura 4: Vecinos vectoriales de movimiento espacial con respecto a una unidad de codificación. Cabe destacar que son los mismos que los usados en las modalidades de FUSIÓN/AMVP en la HEVC. Por lo tanto, no se requiere ningún acceso adicional a la memoria.
- 10 **[0076]** Para verificar bloques de vecindad temporal, el decodificador de vídeo 30 realiza un proceso de construcción de una lista de imágenes candidatas. Hasta dos imágenes de referencia provenientes de la vista actual se pueden tratar como imágenes candidatas. Una imagen de referencia cosituada se inserta primero en la lista de imágenes candidatas, seguida del resto de las imágenes candidatas en orden ascendente del índice de referencia. Cuando las imágenes de referencia con el mismo índice de referencia en ambas listas de imágenes de referencia están disponibles, la que está en la misma lista de imágenes de referencia de la imagen de cosituada precede a la otra. Para
- 15 **[0077]** Cuando un bloque está codificado con predicción de movimiento entre vistas, el decodificador de vídeo 30 obtiene un vector de disparidad para seleccionar un bloque correspondiente en una vista diferente. Un vector de disparidad implícita (IDV, también conocido como vector de disparidad obtenido) se refiere a un vector de disparidad obtenido en la predicción de movimiento entre vistas. Incluso aunque el bloque esté codificado con predicción de movimiento, el vector de disparidad obtenido no se descarta para el fin de codificar un bloque siguiente.
- 20 **[0078]** En el diseño actual de 3D-HTM 7.0 y versiones posteriores de 3D-HTM, el proceso de NBDV verifica los vectores de movimiento de disparidad en los bloques de vecindad temporal, los vectores de movimiento de disparidad en los bloques de vecindad espacial y luego los IDV en orden. Una vez que se encuentra el vector de movimiento de disparidad, o IDV, se termina el proceso. Además, el número de bloques de vecindad espacial verificados en el proceso de NBDV se reduce aún más hasta 2.
- 25 **[0079]** El decodificador de vídeo 30 también puede realizar un refinamiento de la NBDV (NBDV-R) con acceso a información de profundidad. Cuando un vector de disparidad se obtiene a partir del proceso de NBDV, se refina aún más recuperando los datos de profundidad del mapa de profundidades de la vista de referencia. El proceso de refinamiento incluye dos etapas. Primero, el decodificador de vídeo 30 ubica un correspondiente bloque de profundidad por medio del vector de disparidad obtenido en la vista de profundidad de referencia codificada previamente, tal como la vista base. El tamaño del correspondiente bloque de profundidad puede ser el mismo que el de la PU actual. El decodificador de vídeo 30 selecciona luego un valor de profundidad a partir de cuatro píxeles de esquina del correspondiente bloque de profundidad y lo convierte en el componente horizontal del vector de disparidad refinado. La componente vertical del vector de disparidad no se modifica.
- 30 **[0080]** En algunas implementaciones, el vector de disparidad refinado puede usarse, por ejemplo, para la predicción de movimiento entre vistas, mientras que el vector de disparidad sin refinar puede usarse para la predicción residual entre vistas. Además, el vector de disparidad refinado puede almacenarse como el vector de movimiento de una PU si se codifica con la modalidad de predicción de síntesis de retrovisión. En algunas implementaciones, siempre se accederá al componente de vista de profundidad de una vista base, independientemente del valor del índice de orden
- 35 **[0081]** El decodificador de vídeo 30 también se puede configurar para realizar ARP, que es una herramienta de codificación que explota la correlación residual entre las vistas. 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. Actualmente, la ARP solo podría aplicarse a las CU intercodificadas con modalidad de partición igual a Part_2Nx2N. La ARP se aplica tanto al componente de luma (Y) como al componente de croma (Cb y Cr). En la siguiente descripción, la operación (tal como suma, resta) en un bloque (o píxel) significa la operación en cada componente (Y, Cb y Cr) de cada píxel en el
- 40 **[0082]** La figura 5 muestra una estructura ejemplar de predicción para la ARP temporal en la 3D-HEVC, que fue adoptada en la 4ª conferencia de JCT3V, como se propone en el documento JCT3V-D0177. La figura 5 ilustra la estructura de predicción de la ARP para el residual temporal (es decir, la imagen de referencia actual en una lista de imágenes de referencia es una imagen de referencia temporal) en la codificación de vídeo de vista múltiple.
- 45 **[0083]** Como se muestra en la figura 5, el decodificador de vídeo 30 identifica los siguientes bloques en la predicción del bloque actual que se está codificando. El bloque actual se muestra en la figura 5 como Actual 150. La base 151 representa un bloque de referencia en una vista de referencia / base obtenida por el vector de disparidad (DV 152A).
- 50
- 55
- 60
- 65

ActualTRef 153 representa un bloque en la misma vista que el bloque Actual 150 obtenido por un vector de movimiento temporal (TMV 154A) del bloque actual. BaseTRef 155 representa un bloque en la misma vista que el bloque Base 151 obtenido por el vector de movimiento temporal del bloque actual (TMV 154B). Por lo tanto, TMV 154A y TMV 154B corresponden al mismo vector de movimiento, lo que significa que identifican la misma magnitud de desplazamiento a lo largo del eje x y del eje y. La diferencia en la ubicación relativa entre BaseTRef 155 y Actual 150 se puede expresar con un vector de TMV + DV. La diferencia en la ubicación relativa entre ActualTRef 153 y BaseTRef 155 puede expresarse mediante el vector de disparidad DV 152B. TMV + DV y DV 152B se proporcionan en la figura 5 para mostrar la relación entre los diversos bloques y no necesariamente corresponden a los vectores obtenidos o utilizados por el decodificador de vídeo 30.

[0084] Cuando se realiza la ARP temporal, el decodificador de vídeo 30 puede calcular el predictor residual como $\text{BaseTRef} - \text{Base}$, donde la operación de resta se aplica a cada píxel de las formaciones de píxeles indicadas. El decodificador de vídeo 30 puede multiplicar el predictor residual por un factor de ponderación (w). Por lo tanto, el predictor final del bloque actual determinado por el decodificador de vídeo 30 se indica como $\text{ActualTRef} + w * (\text{Base} - \text{BaseTRef})$.

[0085] El ejemplo de la figura 5 muestra el caso de la predicción unidireccional. Cuando se extiende al caso de la predicción bidireccional, el decodificador de vídeo 30 puede aplicar las etapas anteriores para cada lista de imágenes de referencia. Por lo tanto, para la predicción bidireccional, el decodificador de vídeo 30 puede determinar dos predictores residuales para dos bloques predictivos diferentes.

[0086] La figura 6 muestra una relación ejemplar entre un bloque actual 160, un bloque correspondiente 161 y un bloque compensado por movimiento 162. El decodificador de vídeo 30 puede realizar la ARP obteniendo primero un vector de disparidad (DV 163) que apunta a una vista de referencia de destino (V_0). El decodificador de vídeo 30 puede obtener el DV 163 utilizando, por ejemplo, cualquiera de las técnicas especificadas en la 3D-HEVC actual. En la imagen de la vista de referencia V_0 dentro de la misma unidad de acceso, el decodificador de vídeo 30 puede ubicar el bloque correspondiente 161 utilizando el DV 163. El decodificador de vídeo 30 puede reutilizar la información de movimiento del bloque actual 160 para obtener información de movimiento para el bloque de referencia 161. Por ejemplo, si el decodificador de vídeo 30 usó el vector de movimiento 164A, se usó para predecir el bloque actual 160, entonces el decodificador de vídeo 30 puede usar el vector de movimiento 164B para predecir el correspondiente bloque 161. El vector de movimiento 164A y el vector de movimiento 164B están concebidos para representar dos instancias diferentes del mismo vector de movimiento.

[0087] El decodificador de vídeo 30 puede aplicar la compensación de movimiento para el correspondiente bloque 161 basándose en el mismo vector de movimiento usado para codificar el bloque actual 160 y la imagen de referencia obtenida en la vista de referencia para el bloque de referencia, para obtener un bloque residual. El decodificador de vídeo 30 selecciona la imagen de referencia en la vista de referencia (V_0) que tiene el mismo valor de POC (Recuento de Orden de Imagen) que la imagen de referencia de la vista actual (V_m) como la imagen de referencia del bloque correspondiente. El decodificador de vídeo 30 aplica el factor de ponderación al bloque residual para obtener un bloque residual ponderado y agregar los valores del bloque residual ponderado a las muestras predichas.

[0088] El decodificador de vídeo 30 también se puede configurar para realizar la ARP entre vistas. De manera similar a la ARP temporal, cuando una unidad de predicción actual utiliza una imagen de referencia entre vistas, se habilita la predicción de residuos entre vistas. Primero, se calcula el residuo entre vistas dentro de una unidad de acceso diferente, luego la información residual calculada se puede usar para predecir el residuo entre vistas del bloque actual. Esta técnica se propuso en el documento JCT3V-F0123_ y se ha adoptado en la 3D-HEVC.

[0089] La figura 7 muestra una estructura ejemplar de predicción para la ARP entre vistas. Como se muestra en la figura 7, para la ARP entre vistas, el decodificador de vídeo 30 identifica tres bloques relacionados para el bloque actual 170. La base 171 representa el bloque de referencia en la vista de referencia ubicada por el vector de movimiento de disparidad (DMV 172A) del bloque actual 170. BaseRef 173 representa el bloque de referencia de la Base 171 en la vista de referencia ubicada por el vector de movimiento temporal mvLX 174A y el índice de referencia, si está disponible, contenido por la Base 171. ActualRef 175 representa un bloque de referencia en la vista actual identificado al reutilizar la información de movimiento temporal de la Base 171. Por lo tanto, el decodificador de vídeo 30 puede ubicar ActualRef 175 utilizando mvLX 174B, donde mvLX 174A y mvLX 174B representan dos instancias del mismo vector de movimiento. El DMV 172B es igual al DMV 172A, según se incluye en la figura 7 para ilustrar que la disparidad entre Actual 170 y Base 171 es igual a la disparidad entre ActualRef 175 y BaseRef 173. El DMV 172B puede no corresponder realmente a un vector de movimiento de disparidad utilizado o generado por el decodificador de vídeo 30.

[0090] Con los tres bloques identificados, el decodificador de vídeo 30 puede calcular el predictor residual de la señal residual para la PU actual (es decir, Actual 170) como la diferencia entre ActualRef y BaseRef. Además, el predictor entre vistas se puede multiplicar por un factor de ponderación (w). Por lo tanto, el predictor final del bloque actual (Actual 170) determinado por el decodificador de vídeo 30 se indica como $\text{Base} + w * (\text{ActualRef} - \text{BaseRef})$.

[0091] El decodificador de vídeo 30 puede usar el filtrado bilineal para generar los tres bloques relativos, como en algunos diseños conocidos de la ARP para la predicción residual temporal. Además, cuando el vector de movimiento temporal contenido por la Base 171 apunta a una imagen de referencia que se encuentra en una unidad de acceso diferente de la primera imagen de referencia temporal disponible de la PU actual, el decodificador de vídeo 30 puede ajustar a escala el vector de movimiento temporal hasta la primera imagen de referencia temporal disponible y el vector de movimiento ajustado a escala se puede usar para ubicar dos bloques en una unidad de acceso diferente.

[0092] Cuando se aplica la ARP para el residuo entre vistas, la PU actual está utilizando la ARP entre vistas, cuando se aplica la ARP para el residuo temporal, la PU actual está utilizando la ARP temporal.

[0093] En la siguiente descripción, si la referencia correspondiente para una lista de imágenes de referencia es una imagen de referencia temporal y se aplica la ARP, se denota como ARP temporal. De lo contrario, si la referencia correspondiente para una lista de imágenes de referencia es una imagen de referencia entre vistas y se aplica la ARP, se denota como ARP entre vistas.

[0094] Como se ha presentado anteriormente, el decodificador de vídeo 30 puede multiplicar el predictor residual por un factor de ponderación. Normalmente se usan tres factores de ponderación en la ARP (es decir, 0; 0,5 y 1), aunque también se pueden usar más o menos factores de ponderación, así como diferentes factores de ponderación. El codificador de vídeo 20 puede, por ejemplo, seleccionar el factor de ponderación que conduce a un coste mínimo de tasa-distorsión para la CU actual como el factor de ponderación final y señalar el índice del factor de ponderación correspondiente (0, 1 y 2, que corresponden, respectivamente, al factor de ponderación 0; 1 y 0,5) en el flujo de bits al nivel de CU. Todas las predicciones de PU en una CU pueden compartir el mismo factor de ponderación. Cuando el factor de ponderación es igual a 0, la ARP no se utiliza para la CU actual.

[0095] El decodificador de vídeo 30 puede configurarse para realizar una selección de imagen de referencia mediante el ajuste a escala del vector de movimiento. En el documento JCT3V-C0049, las imágenes de referencia de las unidades de predicción codificadas con factores de ponderación distintos de cero pueden ser diferentes de un bloque a otro. Por lo tanto, puede ser necesario acceder a diferentes imágenes desde la vista de referencia para generar el bloque compensado por movimiento (es decir, *BaseTRef* en la Figura 5) del bloque correspondiente. Cuando el factor de ponderación no es igual a 0, para el residuo 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 de residuo y de predictor residual. Cuando se aplica la ARP al residuo entre vistas, los vectores de movimiento temporal del bloque de referencia (es decir, la *Base* en la Figura 7) se ajustan a escala hacia una imagen fija antes de realizar la compensación de movimiento para ambos procesos de generación de residuo y de predictor residual.

[0096] Para ambos casos (es decir, residuo temporal o residuo entre vistas), 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 decodificado no apunta a la imagen fija, primero se ajusta a escala y luego se usa para identificar *ActualTRef* y *BaseTRef*.

[0097] Dicha imagen de referencia utilizada para la ARP se denomina imagen de referencia de ARP de destino. Téngase en cuenta que cuando el fragmento actual es un fragmento B, la imagen de referencia de ARP de destino está asociada a la lista de imágenes de referencia. Por lo tanto, se pueden utilizar dos imágenes de referencia de ARP de destino.

[0098] El decodificador de vídeo 30 puede realizar una verificación de disponibilidad de las imágenes de referencia de ARP de destino. *RpRefImgLX* puede indicar la imagen de referencia de ARP de destino asociada a una lista X de imágenes de referencia (con X igual a 0 o 1), y la imagen en la vista con índice de orden de vista igual al obtenido del proceso de NBDV y con el mismo valor de POC de *RpRefImgLX* puede ser indicada por *RefImgInRefVistaLX*. Cuando una de las siguientes condiciones es falsa, el decodificador de vídeo 30 puede inhabilitar la ARP, inhabilitada para la lista X de imágenes de referencia: (1) *RpRefImgLX* no está disponible, (2) *RefImgInRefVistaLX* no se almacena en el almacén temporal de imágenes decodificadas, (3) *RefImgInRefVistaLX* no está incluida en ninguna de las listas de imágenes de referencia del bloque correspondiente (es decir, *Base* en la Figura 5 y la Figura 7) localizado por el DV desde el proceso de NBDV o el DMV asociado al bloque actual, y la ARP puede estar inhabilitada para esta lista de imágenes de referencia.

[0099] Cuando se aplica la ARP, el decodificador de vídeo 30 puede usar un filtro bilineal al generar el residuo y el predictor residual. Es decir, los tres bloques que excluyen el bloque actual implicado en el proceso de ARP pueden generarse utilizando un filtro bilineal.

[0100] El decodificador de vídeo 30 también puede realizar la ARP a nivel de bloque. A diferencia de la descripción anterior donde todos los bloques dentro de una PU comparten la misma información de movimiento, a veces denominada ARP a nivel de PU, en la ARP a nivel de bloque, el decodificador de vídeo 30 divide una PU en varios bloques de tamaño 8x8, y cada bloque de tamaño 8x8 tiene su propia información de movimiento para realizar la ARP. Cuando se habilita la ARP a nivel de bloque, ya sea temporal o entre vistas, cada PU se divide primero en varios

bloques, y cada bloque comparte la misma información de movimiento que la PU actual. Sin embargo, el vector de movimiento obtenido (es decir, el vector de disparidad en la ARP temporal o el vector de movimiento temporal en la ARP entre vistas) puede actualizarse para cada bloque de tamaño 8x8.

5 **[0101]** La figura 8A muestra una estructura ejemplar de predicción de la ARP temporal a nivel de bloque. En el ejemplo de la figura 8A, Actual 180 representa una PU que se divide en cuatro bloques de tamaño 8x8, etiquetados A a D en la figura 8A. La base 181 representa cuatro bloques de referencia (etiquetados A' a D') en una vista de referencia / base obtenida por los vectores de disparidad de Actual 180. El bloque A' de la Base 181 se identifica utilizando el vector de disparidad del bloque A (mostrado en la Figura 8A como DV[0]), y el Bloque B' se identifica utilizando el vector de disparidad (mostrado en la Figura 8A como DV[1]). Aunque no se muestra explícitamente en la figura 8A, los bloques C' y D' también pueden identificarse utilizando los vectores de disparidad de los bloques C y D.

15 **[0102]** El vector de movimiento obtenido (es decir, el vector de disparidad en la ARP temporal) puede actualizarse para cada bloque de tamaño 8x8. Para la ARP temporal, el Mv obtenido por omisión (indicado por DV[i] para el i-ésimo bloque de tamaño 8x8 en la Figura 8A) se fija primero para que sea el DV de un proceso de NBDV. Cuando el bloque que cubre la posición central del i-ésimo bloque de tamaño 8x8 dentro de ActualRef contiene un vector de movimiento de disparidad, DV[i] se actualiza para que sea ese vector de movimiento de disparidad. Por lo tanto, como se muestra en la figura 8A, los bloques A' a D' se pueden situar entre sí de manera diferente a como lo están los bloques A a D entre sí. ActualRef 183 representa a cuatro bloques (A_P a D_P) en la misma vista que Actual 180, obtenido por el vector de movimiento temporal (mostrado como mvLX 184A en la Figura 8A) de Actual 180. BaseRef 185 representa a los cuatro bloques (A_R a D_R) en la misma vista que Base 181, obtenido por el vector de movimiento temporal del bloque actual (mvLX 184B). En el ejemplo de la figura 8A, mvLX 184A y mvLX 184B pretenden representar dos aplicaciones diferentes del mismo vector de movimiento. Es decir, mvLX 184A y mvLX 184B tienen los mismos componente x y componente y.

25 **[0103]** El predictor residual, en el ejemplo de la figura 8A, se indica como BaseRef-Base, donde la operación de resta se aplica a cada píxel de las formaciones de píxeles indicados. Un factor de ponderación (w) se multiplica adicionalmente por el predictor residual. Por lo tanto, los predictores finales para los bloques A a D determinados por el decodificador de vídeo 30 se indican como ActualRef [N_P]+w*(Base[N']-BaseRef [N_R]), con N correspondiente a A a D.

35 **[0104]** La figura 8B muestra una estructura ejemplar de predicción de la ARP entre vistas a nivel de bloque. En el ejemplo de la figura 8B, el decodificador de vídeo 30 identifica tres bloques relacionados del bloque actual 182. La base 186 representa los cuatro bloques de referencia (A a D) en la vista de referencia ubicada por el vector de movimiento de disparidad (DMV 188A) del bloque actual 182. BaseRef 187 representa los cuatro bloques de referencia (A' a D') de la Base 186 en la vista de referencia ubicada por el vector de movimiento temporal mvLX[N] y el índice de referencia, si está disponible, contenido por la Base 186, donde N corresponde a los bloques A D. Para la ARP entre vistas, el Mv obtenido por omisión (indicado por mvLX[i] para el i-ésimo bloque de tamaño 8x8 en la Figura 8B) se puede fijar en el vector de movimiento temporal asociado a un bloque que cubre la posición central de Base, como en la ARP actual. Cuando el bloque que cubre la posición central del i-ésimo bloque de tamaño 8x8 dentro de Base contiene un vector de movimiento temporal, mvLX[i] se actualiza para que sea ese vector de movimiento temporal. Por lo tanto, como se muestra en la figura 8A, los bloques A' a D' se pueden situar entre sí de manera diferente a como lo están los bloques A a D entre sí.

45 **[0105]** ActualRef 189 representa a cuatro bloques de referencia (A_R - D_R) en la vista actual identificada al reutilizar la información de movimiento temporal desde la Base 186. Así, por ejemplo, el decodificador de vídeo 30 localiza A_R usando mvLX[A], ubica B_R usando mvLX[B], y así sucesivamente. Con los tres bloques identificados, el decodificador de vídeo 30 puede calcular el predictor residual de la señal residual de la PU actual como la diferencia entre ActualRef - BaseRef. Que pueden estar en diferentes unidades de acceso. Además, el predictor entre vistas se puede multiplicar por un factor de ponderación (w). Por lo tanto, el predictor final del bloque actual determinado por el decodificador de vídeo 30 se indica como Base[N]+w*(ActualRef[N_R]-BaseRef[N']).

50 **[0106]** Como se ha ilustrado anteriormente, tanto para la ARP temporal basada en bloques como para la ARP entre vistas basada en bloques, solo se accede a la información de movimiento a nivel de bloque (por ejemplo, de tamaño 8x8) del bloque de referencia localizado por el vector de movimiento de la PU actual para generar el predictor residual final.

55 **[0107]** El decodificador de vídeo 30 también puede realizar la predicción de movimiento entre vistas a nivel de sub-PU. En el documento JCT3V-F0110 se propone un procedimiento de predicción de movimiento entre vistas a nivel de sub-PU para generar un nuevo candidato de fusión. El nuevo candidato se agrega a la lista de candidatos de fusión. El nuevo candidato, denominado candidato de fusión de sub-PU, puede obtenerse mediante el decodificador de vídeo 30 usando el siguiente procedimiento. En la siguiente descripción, el tamaño de una PU actual se indica por nPSW x nPSH, el tamaño de sub-PU señalado por NxN y el tamaño de sub-PU final por subWxsubH. El decodificador de vídeo 30 divide primero la PU actual en una o varias sub-PU en función del tamaño de PU y el tamaño de sub-PU señalado.

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

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

5 **[0108]** El decodificador de vídeo 30 en segundo lugar establece el vector de movimiento predeterminado tmvLX en (0, 0) y el índice de referencia refLX en -1 para cada lista de imágenes de referencia (siendo X 0 y 1). Para cada sub-PU en el orden de barrido por trama, el decodificador de vídeo 30 hace lo siguiente:

- 10 ○ agregar el DV del proceso de DoNBDV o NBDV a la posición intermedia de la sub-PU actual para obtener una ubicación de muestra de referencia (xRefSub , yRefSub) de la siguiente manera:

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

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

15 un bloque en la vista de referencia que abarca (xRefSub , yRefSub) puede usarse como el bloque de referencia para la sub-PU actual.

- 20 ○ Para el bloque de referencia identificado,

- si se codifica utilizando vectores de movimiento temporal, vale lo siguiente:

- 25 • 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.
- 30 - De lo contrario (el bloque de referencia está intracodificado), la información de movimiento de la sub-PU actual se puede establecer en tmvLX y refLX .

35 **[0109]** El decodificador de vídeo 30 también se puede configurar para realizar la ARP a nivel de sub-PU. Cuando se aplica la predicción de movimiento entre vistas a nivel de sub-PU, una PU puede contener múltiples sub-PU y cada sub-PU tiene su propia información de movimiento, y la ARP puede realizarse para cada sub-PU. Se pueden aplicar diferentes tamaños de bloques de sub-PU, por ejemplo, 4x4, 8x8 y 16x16. El tamaño del bloque de sub-PU está presente en el conjunto de parámetros de vista.

40 **[0110]** La figura 9 muestra un ejemplo de predicción de movimiento entre vistas al nivel de sub-PU. La figura 9 muestra una vista actual, denominada VI, y una vista de referencia, denominada V0. La PU actual 190 incluye cuatro sub-PU A a D. El decodificador de vídeo 30 puede usar vectores de disparidad de cada una de las cuatro sub-PU A a D para ubicar los bloques de referencia 191, que incluyen cuatro bloques de referencia A_R a D_R . Los vectores de disparidad de las sub-PU A a D se muestran en la figura 9 como $\text{MV}[i]$, donde i corresponde a A a D. Como cada una de las cuatro sub-PU tiene un vector de disparidad único, la ubicación de las sub-PU A a D entre sí puede diferir de la ubicación de los bloques de referencia A_R a D_R entre sí. En la predicción de movimiento entre vistas al nivel de sub-PU, el decodificador de vídeo 30 puede usar el vector de movimiento de un bloque de referencia para predecir una sub-PU. Los vectores de movimiento de los bloques de referencia A_R a D_R se muestran en la figura 9 como $\text{MV}[i]$, donde i corresponde a A a D. Por lo tanto, como ejemplo, para la sub-PU A, el decodificador de vídeo 30 puede usar $\text{DV}[A]$ para ubicar el bloque de referencia A_R , determinar que el bloque de referencia A_R se codificó usando $\text{MV}[A]$ y usar $\text{MV}[A]$ para ubicar un bloque predictivo para la sub-PU A.

55 **[0111]** La figura 10A muestra una estructura de predicción ejemplar para la ARP temporal al nivel de sub-PU. En el ejemplo de la figura 10A, una PU (Actual 200) se divide en cuatro sub-PU (etiquetadas A a D en la figura 10A). Para la ARP temporal al nivel de sub-PU, el decodificador de vídeo 30 puede usar el mismo vector de disparidad (DV 202) para todas las sub-PU de Actual 200, para identificar el bloque de referencia (Base 201) en la vista de referencia, que generalmente es el mismo que en la ARP al nivel de PU. La base 201 puede subdividirse en subbloques de referencia (A' a D' en la Figura 10A) que corresponden a las sub-PU A a D. El decodificador de vídeo 30 puede obtener el DV 202, por ejemplo, utilizando una técnica de NBDV. El decodificador de vídeo 30 usa información de movimiento de cada una de las sub-PU A a D para identificar bloques de referencia temporales (A_P a D_P en la Figura 10A). La información de movimiento de las sub-PU A a D se muestra en la figura 10A como $\text{TMV}[i]$ para la i -ésima sub-PU,

donde i corresponde a A a D. El $TMV[A]$, por ejemplo, representa el vector de movimiento temporal de la sub-PU A y el $TMV[C]$ representa el vector de movimiento de la sub-PU C. Aunque no se muestra explícitamente en la figura 10A, la sub-PU B y la sub-PU D tendrían, de manera similar, vectores de movimiento asociados, $TMV[B]$ y $TMV[D]$, respectivamente.

5
[0112] El decodificador de vídeo 30 puede reutilizar la información de movimiento de las sub-PU A a D (es decir, los $TMV[i]$, para $i = A$ a D) para ubicar los bloques de referencia de la Base 201, mostrados como BaseRef 205 en la figura 10A. BaseRef 205 incluye cuatro subbloques (A_R a D_R en la Figura 10A). El predictor residual, en el ejemplo de la figura 10A, se puede indicar como BaseRef-Base, donde la operación de resta se aplica a cada píxel de las formaciones de píxeles indicadas. Un factor de ponderación (w) se multiplica adicionalmente por el predictor residual. Por lo tanto, el predictor final para los bloques A a D, determinado por el decodificador de vídeo 30, se puede indicar como $ActualRef[N_P] + w * (Base[N'] - BaseRef[N_R])$, con N correspondiente a A a D.

15
[0113] La figura 10B muestra una estructura de predicción ejemplar de la ARP entre vistas a nivel de sub-PU. En el ejemplo de la figura 10B, una PU (Actual 200) se divide en cuatro sub-PU (etiquetadas A a D en la figura 10B). Para la ARP entre vistas, el decodificador de vídeo 30 usa un vector de movimiento de disparidad de cada una de las sub-PU A a D, para identificar un bloque de referencia (Base 206) en una vista de referencia. La base 206 incluye cuatro subbloques de referencia, etiquetados A_P a D_P en la figura 10B. El vector de movimiento de disparidad de las sub-PU A a D se muestra en la figura 10B como $DMV[i]$ para la i -ésima sub-PU, donde i corresponde a A a D. $DMV[A]$, por ejemplo, representa el vector de movimiento de disparidad de la sub-PU A y $DMV[B]$ representa el vector de movimiento de disparidad de la sub-PU B. Aunque no se muestra explícitamente en la figura 10B, la sub-PU C y la sub-PU D tendrían, de manera similar vectores asociados de movimiento de disparidad, $DMV[C]$ y $DMV[D]$, respectivamente.

25
[0114] Cuando el bloque de referencia (es decir, la Base 206) contiene un vector de movimiento temporal (indicado por $mvLX[i]$ en la Figura 10B, donde i corresponde a A a D), el decodificador de vídeo 30 usa el vector de movimiento temporal para identificar un bloque de referencia temporal, tanto para la sub-PU actual como para su bloque de referencia en la vista de referencia. Por ejemplo, el decodificador de vídeo 30 usa $mvLX[A]$ para ubicar un bloque de referencia para A_P , que es A_R en la figura 10B, así como para localizar un bloque de referencia de A, que es A' en la figura 10B. El decodificador de vídeo 30 puede usar, de manera similar, $mvLX[C]$ para ubicar un bloque de referencia para C_P , que es C_R en la figura 10B, así como para localizar un bloque de referencia de C, que es C' en la figura 10B. Aunque no se muestra explícitamente en la figura 10B, el decodificador de vídeo 30, de manera similar, puede ubicar bloques de referencia para C, C_P , D y D_P .

30
[0115] Con los bloques identificados, el decodificador de vídeo 30 puede calcular el predictor residual de la PU actual como la diferencia entre $ActualRef[N'] - BaseRef[N_R]$, donde N corresponde a A a D. Además, el predictor entre vistas se puede multiplicar por un factor de ponderación (w). Por lo tanto, el predictor final del bloque actual determinado por el decodificador de vídeo 30 se puede indicar como $Base[N_P] + w * (ActualRef[N'] - BaseRef[N_R])$.

35
[0116] Algunas implementaciones de la ARP tienen algunos problemas potenciales. Como ejemplo, en algunos escenarios de codificación donde un bloque es bi-predicho, es posible que se deban evaluar cuatro bloques de referencia adicionales para un bloque (o PU, sub-PU). En un primer ejemplo, ilustrado por la figura 11, cuando un bloque se predice bidireccionalmente y ambas direcciones de predicción corresponden a imágenes de referencia entre vistas, se invoca la ARP entre vistas dos veces y se accede a dos bloques de referencia adicionales para cada ARP.

40
[0117] La figura 11 muestra un ejemplo de bloques de referencia a los que se accede mediante el decodificador de vídeo 30 para la ARP entre vistas bidireccional en la 3D-HEVC. En el ejemplo de la figura 11, el vector de movimiento de disparidad de la dirección de predicción X se indica por $DMVX$, donde $X = 0$ o 1. Para la dirección de predicción X, se identifica un bloque de referencia en la vista actual ($ActualRefX$ en la Figura 11) mediante información de movimiento ($mvBaseX$ en la Figura 11) asociada al bloque de referencia en la vista de referencia ($BaseX$ en la Figura 11), y se evalúa un bloque de referencia de $BaseX$ en la vista de referencia ($BaseXRef$ en la Figura 11) identificada por $DMVX + mvBaseX$.

45
[0118] La figura 12 muestra un ejemplo de bloques de referencia a los que se accede mediante el decodificador de vídeo 30 para la ARP temporal y la ARP entre vistas en la 3D-HEVC. En un segundo ejemplo, ilustrado por la figura 12, cuando un bloque se predice de manera bidireccional, y una dirección de predicción corresponde a una imagen de referencia temporal (y el vector de movimiento temporal es TMV) y la otra dirección de predicción corresponde a una imagen de referencia entre vistas (y el vector de movimiento de disparidad es DMV), se invocan tanto la ARP temporal como la ARP entre vistas, y se accede a dos bloques de referencia adicionales para cada ARP, como se muestra en la figura 12.

50
[0119] En la ARP temporal, se evalúan un bloque de referencia en la vista de referencia (Base1 en la Figura 12) identificada por el DV obtenido utilizando un proceso de NBDV y un bloque de referencia de Base1 en la vista de referencia ($Base1TRef$ en la Figura 12) identificada por $DV + TMV$. En la ARP entre vistas, un bloque de referencia en la vista actual ($ActualRef$ en la Figura 12) identificado por información de movimiento ($mvBase$ en la Figura 12)

asociada al bloque de referencia en la vista de referencia (Base2 en la Figura 12), y un bloque de referencia de Base2 en la vista de referencia (Base2Ref en la Figura 12) identificado por el DMV + mvBase.

5 **[0120]** Según algunas técnicas conocidas, el proceso de la figura 12 se simplifica para reducir los bloques de referencia evaluados adicionalmente. Por ejemplo, se puede usar un DMV en lugar de un DV obtenido utilizando un proceso de NBDV para identificar el bloque de referencia en la vista de referencia (es decir, Basel en la Figura 12) para la ARP temporal. De esta manera, el bloque Basel es el mismo que el bloque Base2 en la figura 12 y no se requieren evaluaciones adicionales de Basel. Por lo tanto, los bloques de referencia evaluados adicionalmente en el primer ejemplo se reducen de cuatro a tres.

10 **[0121]** Sin embargo, en el primer ejemplo del problema anterior, todavía hay cuatro bloques de referencia adicionales para evaluar. Esto hace que el peor caso de la cantidad de bloques necesarios para acceder a un bloque predicho por la ARP se incremente de tres a cuatro.

15 **[0122]** Esta divulgación proporciona potencialmente soluciones para algunos de los problemas mencionados anteriormente en la ARP para reducir los bloques de referencia evaluados adicionalmente. Como ejemplo, se propone que cuando un primer bloque se codifica con ARP (incluida la ARP a nivel de sub-PU) y se predice de manera bidireccional y ambas direcciones de predicción tienen imágenes de referencia que son imágenes de referencia entre vistas, el decodificador de vídeo 30 puede usar un solo vector de movimiento temporal para identificar el bloque de referencia del bloque actual en la vista actual para (la ARP entre vistas de) ambas direcciones de predicción. En otras palabras, ambos vectores de movimiento temporal (por ejemplo, mvBase0 y mvBase1, como se muestran en la Figura 20 11) se fijan para que sean mvBase. Además, solo se determina un bloque de referencia del bloque actual, a diferencia tanto de ActualRef0 como de ActualRef1 en la figura 12. En este caso, solo se evalúa un bloque de referencia (indicado por ActualRef en la Figura 13), en lugar de dos bloques de referencia, en una vista actual, como se muestra en la figura 13.

25 **[0123]** La figura 13 muestra un ejemplo de cómo se puede usar un solo vector de movimiento temporal en la ARP bidireccional entre vistas. En un ejemplo, el único vector de movimiento temporal (mvBase) se puede configurar para que sea el vector de movimiento temporal asociado al bloque de referencia en la vista de referencia para la dirección de predicción 0 (por ejemplo, mvBase0). Además, cuando mvBase0 no está disponible, la ARP puede estar inhabilitada para el primer bloque. Alternativamente, cuando mvBase0 no está disponible, el vector de movimiento único (mvBase) puede configurarse para que sea el vector de movimiento cero.

30 **[0124]** En el ejemplo de la figura 13, el decodificador de vídeo 30 puede realizar la ARP entre vistas para dos direcciones de predicción. Para la dirección de predicción 0, el decodificador de vídeo 30 determina un primer vector de movimiento de disparidad (DMV0) para Actual y un segundo vector de movimiento de disparidad (DMV1) para Actual. El decodificador de vídeo 30 usa el DMV0 para ubicar un primer bloque correspondiente (Base0) y usa el DMV1 para ubicar un segundo bloque correspondiente (Base1). A partir de los vectores de movimiento de Base0 y Basel, el decodificador de vídeo 30 determina un vector de movimiento (mvBase) a usar para la ARP. Los diversos procesos que el decodificador de vídeo 30 puede usar para determinar mvBase se explicarán con mayor detalle a continuación. Usando mvBase, el decodificador de vídeo 30 determina un bloque de referencia (ActualRef) del bloque actual en una imagen diferente en la misma vista que Actual. Usando mvBase, el decodificador de vídeo 30 también determina un bloque de referencia para Base 0 (Base0Ref) y un bloque de referencia para Basel (Base1Ref). Usando los bloques identificados, el decodificador de vídeo 30 genera dos predictores. El primer predictor es $Base0 + w * (ActualRef - Base0Ref)$ y el segundo predictor es $Base1 + w * (ActualRef - Base1Ref)$.

35 **[0125]** El decodificador de vídeo 30 puede determinar que mvBase sea el vector de movimiento temporal asociado a Base0 si está disponible un vector de movimiento para Base0, o puede determinar que mvBase sea el vector de movimiento temporal asociado a Base1 si hay disponible un vector de movimiento para Basel. Si el decodificador de vídeo 30 está configurado para usar el vector de movimiento de Base0 como mvBase, entonces la ARP se puede desactivar para el primer bloque cuando un vector de movimiento para Base0 no esté disponible. Alternativamente, si el decodificador de vídeo 30 está configurado para usar el vector de movimiento de Base0 como mvBase, entonces se puede configurar mvBase para que sea el vector de movimiento cero cuando un vector de movimiento de Base0 no esté disponible. De manera similar, si el decodificador de vídeo 30 está configurado para usar el vector de movimiento de Basel como mvBase, entonces la ARP puede inhabilitarse para el primer bloque cuando un vector de movimiento para Basel no esté disponible. Alternativamente, si el decodificador de vídeo 30 está configurado para usar el vector de movimiento de Base1 como mvBase, entonces se puede configurar mvBase para que sea el vector de movimiento cero cuando un vector de movimiento de Basel no esté disponible.

40 **[0126]** En otro ejemplo, el decodificador de vídeo 30 puede configurar mvBase para que sea el vector de movimiento temporal de Base1 si un vector de movimiento para Base0 no está disponible, o puede configurar mvBase para que sea el vector de movimiento temporal de Base0 si un vector de movimiento para Base1 no está disponible. El decodificador de vídeo puede configurar mvBase para que sea un vector de movimiento cero si un vector de movimiento para Base1 no está disponible y si un vector de movimiento para Base0 no está disponible. El decodificador de vídeo puede inhabilitar la ARP si un vector de movimiento para Base1 no está disponible y si un vector de movimiento para Base0 no está disponible. En otro ejemplo, el decodificador de vídeo 30 puede inhabilitar la ARP

entre vistas para la dirección de predicción X, cuando el vector de movimiento temporal asociado al bloque de referencia en la vista de referencia no está disponible para la dirección de predicción X.

[0127] De acuerdo a otra técnica de esta divulgación, se propone que cuando un bloque se codifica con ARP (incluida la ARP a nivel de sub-PU) y se predice de manera bidireccional, el decodificador de vídeo 30 aplica la ARP de croma solo para una dirección de predicción (dirección de predicción X) y desactiva la ARP para la otra dirección de predicción (dirección de predicción 1 - X), donde X puede ser 0 o 1. La ARP de luma (incluida la ARP a nivel de sub-PU) puede mantenerse sin cambios. En un ejemplo, X es igual a 0. El decodificador de vídeo 30 puede usar esta técnica, ya sea de manera conjunta o independiente de la técnica de vector de movimiento único descrita anteriormente.

[0128] De acuerdo a otra técnica de esta divulgación, se propone que, cuando un bloque se codifica con ARP, además se aplica la ARP para componentes de croma solo cuando el tamaño del bloque está en un cierto rango, lo que significa que el ancho y la altura del bloque actual están en un cierto rango. En un ejemplo, si un tamaño de bloque es igual a 8×8 , la ARP para componentes de croma puede ser inhabilitada. En otro ejemplo, si el tamaño de un bloque es menor que 32×32 , la ARP para componentes de croma puede ser inhabilitada. En otro ejemplo, la ARP a nivel de sub-PU para croma puede estar inhabilitada para cualquier sub-PU con un tamaño igual a $N \times N$, pero la ARP para croma está habilitada para una PU con un tamaño igual a $N \times N$. Aquí, N puede ser 8, 16, 32 o 64. En otro ejemplo, la ARP a nivel de sub-PU para croma puede estar inhabilitada para cualquier sub-PU con un tamaño igual a $N \times N$, pero la ARP para croma está habilitada para una PU con un tamaño igual a $M \times M$. Aquí, M puede ser más pequeño que N y ambos pueden ser 8, 16, 32 o 64 siempre que M sea más pequeño que N.

[0129] Se contempla que las diversas técnicas descritas en esta divulgación pueden implementarse de forma independiente o conjunta. Por ejemplo, las técnicas de vectores de movimiento únicos, descritas anteriormente pueden implementarse junto con las técnicas de ARP de croma descritas anteriormente. De manera similar, también se contempla que las técnicas de ARP de croma basadas en el tamaño de bloque, descritas anteriormente, se pueden implementar junto con las técnicas de vectores de movimiento únicos, descritas anteriormente. Además, se contempla que las diversas técnicas descritas en esta divulgación se pueden aplicar a cualquiera entre la ARP a nivel de PU, la ARP a nivel de sub-PU y la ARP a nivel de bloque.

[0130] La figura 14 es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo que puede implementar las técnicas de ARP descritas en esta divulgación. Por ejemplo, la figura 14 ilustra el codificador de vídeo 20 que puede representar un codificador de vídeo conforme a la 3D-AVC o a la 3D-HEVC. El codificador de vídeo 20 se describirá usando cierta terminología de la HEVC, tal como PU, TU y CU, pero debería entenderse que las técnicas descritas con referencia al codificador de vídeo 20 también pueden realizarse con vídeo codificado de acuerdo a la norma H.264.

[0131] El codificador de vídeo 20 puede realizar la intracodificación y la intercodificación de bloques de vídeo dentro de fragmentos de vídeo. Por ejemplo, el codificador de vídeo 20 puede realizar la codificación de interpredicción o la codificación de intrapredicción. 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 dadas. La inter-codificación se basa en la predicción temporal o en la predicción entre vistas para reducir o eliminar la redundancia temporal dentro de tramas o imágenes adyacentes de una secuencia de vídeo o la redundancia entre imágenes en diferentes vistas. La intramodalidad (modalidad I) puede referirse a cualquiera entre varias modalidades de compresión de base espacial. Las intermodalidades, tales como la predicción unidireccional (modalidad P) o la predicción bidireccional (modalidad B), pueden referirse a cualquiera entre varias modalidades de compresión de base temporal.

[0132] En el ejemplo de la figura 14, el codificador de vídeo 20 incluye una memoria de datos de vídeo 40, una unidad de procesamiento de predicciones 42, una memoria de imágenes de referencia 64, un sumador 50, una unidad de procesamiento de transformaciones 52, una unidad de procesamiento de cuantización 54 y una unidad de codificación por entropía 56. La unidad de procesamiento de predicción 42 incluye la unidad de estimación de movimiento y disparidad 44, la unidad de compensación de movimiento y disparidad 46 y la unidad de intrapredicción 48. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 también incluye una unidad de procesamiento de cuantización inversa 58, una unidad de procesamiento de transformaciones inversas 60 y un sumador 62. También se puede incluir un filtro de eliminación de bloques (no mostrado en la figura 14) para filtrar fronteras de bloques y eliminar distorsiones de efecto pixelado del vídeo reconstruido. Si se desea, el filtro de eliminación de bloques filtrará habitualmente la salida del sumador 62. También se pueden usar filtros de bucle adicionales (en el bucle o tras el bucle), además del filtro de eliminación de bloques.

[0133] La memoria de datos de vídeo 40 puede almacenar datos de vídeo a codificar por 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 del origen de vídeo 18. La memoria de imágenes de referencia 64 es un ejemplo de una memoria temporal de imágenes de decodificación (DPB), 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 las modalidades de intracodificación o inter-codificación, también denominadas modalidades de codificación de intrapredicción o interpredicción). La memoria de datos de vídeo 40 y la memoria de imágenes de referencia 64 pueden ser formadas por cualquiera entre varios dispositivos de memoria, tales como memoria dinámica de acceso aleatorio (DRAM), incluyendo DRAM síncrona

(SDRAM), RAM magnetorresistiva (MRAM), RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. La memoria de datos de vídeo 40 y la memoria de imágenes de referencia 64 pueden ser proporcionadas por el mismo dispositivo de memoria o por dispositivos de memoria independientes. En diversos ejemplos, la memoria de datos de vídeo 40 puede estar en un chip con otros componentes del codificador de vídeo 20, o fuera de chip con respecto a esos componentes.

[0134] El codificador de vídeo 20 recibe datos de vídeo y una unidad de partición (no mostrada) divide los datos en bloques de vídeo. Esta partición también puede incluir la partición en fragmentos, mosaicos u otras unidades más grandes, así como la partición de bloques de vídeo (por ejemplo, particiones de macrobloques y subbloques de particiones). El codificador de vídeo 20 ilustra, en general, los componentes que codifican bloques de vídeo dentro de un fragmento de vídeo a codificar. El fragmento se puede dividir en múltiples bloques de vídeo (y, posiblemente, en conjuntos de bloques de vídeo denominados mosaicos). La unidad de procesamiento de predicción 42 puede seleccionar una entre una pluralidad de posibles modalidades de codificación, tal como una entre una pluralidad de modalidades de intracodificación (modalidades de codificación de intrapredicción) o una entre una pluralidad de modalidades de inter-codificación (modalidades de codificación de interpredicción), para el bloque de vídeo actual, basándose en los resultados de errores (por ejemplo, la tasa de codificación y el nivel de distorsión). La unidad de procesamiento de predicción 42 puede proporcionar el bloque intracodificado o intercodificado resultante al sumador 50 para generar datos de bloques residuales y al sumador 62 para reconstruir el bloque codificado para su uso como una imagen de referencia.

[0135] La unidad de intrapredicción 48, dentro de la unidad de procesamiento de predicción 42, 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 a codificar, para proporcionar compresión espacial. La unidad de estimación de movimiento y disparidad 44 y la unidad de compensación de movimiento y disparidad 46 dentro de la unidad de procesamiento de predicción 42 llevan a cabo la codificación interpredictiva del bloque de vídeo actual en relación con uno o más bloques predictivos en una o más imágenes de referencia para proporcionar compresión temporal.

[0136] La unidad de estimación de movimiento y disparidad 44 puede estar configurada para determinar la modalidad de interpredicción para un fragmento de vídeo de acuerdo a un patrón predeterminado para una secuencia de vídeo. El patrón predeterminado puede designar fragmentos de vídeo en la secuencia como fragmentos P o fragmentos B. La unidad de estimación de movimiento y disparidad 44 y la unidad de compensación de movimiento y disparidad 46 pueden estar sumamente integradas, pero se ilustran por separado con fines conceptuales. La estimación de movimiento, llevada a cabo por la unidad de estimación de movimiento y disparidad 44, es el proceso de generar vectores de movimiento, que estiman el movimiento para los bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de un bloque de vídeo dentro de una trama o imagen de vídeo actual a un bloque predictivo dentro de una imagen de referencia.

[0137] Un bloque predictivo es un bloque que se revela como estrechamente coincidente con el bloque de vídeo a codificar, en lo que respecta a la diferencia de píxeles, lo cual puede determinarse mediante suma de diferencias absolutas (SAD), suma de las diferencias al cuadrado (SSD) u otras métricas de diferencia. En algunos ejemplos, el codificador de vídeo 20 puede calcular los valores para las posiciones fraccionarias de píxeles de imágenes de referencia almacenadas en la memoria de imágenes de referencia 64. Por ejemplo, el codificador de vídeo 20 puede interpolar valores de posiciones de un cuarto de píxel, posiciones de un octavo de píxel u otras posiciones fraccionarias de píxel de la imagen de referencia. Por lo tanto, la unidad de estimación de movimiento y disparidad 44 puede realizar una búsqueda de movimiento en relación con las posiciones de píxeles completas y las posiciones de píxeles fraccionarias, y emitir un vector de movimiento con una precisión de píxel fraccionaria.

[0138] La unidad de estimación de movimiento y disparidad 44 calcula un vector de movimiento para un bloque de vídeo en un fragmento intercodificado (codificado por interpredicción) comparando la posición del bloque de vídeo con la posición de un bloque predictivo de una imagen de referencia. La imagen de referencia se puede seleccionar entre una primera lista de imágenes de referencia (RefImgList0) o una segunda lista de imágenes de referencia (RefImgList1), cada una de las cuales identifica una o más imágenes de referencia almacenadas en la memoria de imágenes de referencia 64. La unidad de estimación de movimiento y disparidad 44 envía el vector de movimiento calculado a la unidad de codificación por entropía 56 y a la unidad de compensación de movimiento y disparidad 46.

[0139] La compensación de movimiento, realizada por la unidad de compensación de movimiento y disparidad 46, puede implicar extraer o generar el bloque predictivo basándose en el vector de movimiento determinado mediante estimación de movimiento, realizando posiblemente interpolaciones hasta una precisión de subpíxel. Tras recibir el vector de movimiento para el bloque de vídeo actual, la unidad de compensación de movimiento y disparidad 46 puede ubicar 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 está codificando, formando valores de diferencia de píxel. Los valores de diferencia de píxel forman datos residuales para el bloque, y pueden incluir componentes de diferencia tanto de luma como de croma. El sumador 50 representa el componente o los componentes que realizan esta operación de resta. La unidad de compensación de movimiento y disparidad 46 también puede generar elementos

sintácticos asociados a los bloques de vídeo y al fragmento de vídeo para su uso por el decodificador de vídeo 30 en la decodificación de los bloques de vídeo del fragmento de vídeo.

5 **[0140]** La unidad de intrapredicción 48 puede intrapredicir un bloque actual, de forma alternativa a la interpredicción llevada a cabo por la unidad de estimación de movimiento y disparidad 44 y la unidad de compensación de movimiento y disparidad 46, como se ha descrito anteriormente. En particular, la unidad de intrapredicción 48 puede determinar una modalidad de intrapredicción a usar para codificar un bloque actual. En algunos ejemplos, la unidad de intrapredicción 48 puede codificar un bloque actual usando diversas modalidades de intrapredicción, por ejemplo, durante pasadas de codificación independientes, y la unidad de intrapredicción 48 (o una unidad de selección de modalidad, en algunos ejemplos) puede seleccionar una modalidad adecuada de intrapredicción a usar a partir de las modalidades probadas. Por ejemplo, la unidad de intrapredicción 48 puede calcular valores de velocidad-distorsión usando un análisis de velocidad-distorsión para las diversas modalidades de intrapredicción probadas, y seleccionar la modalidad de intrapredicción que tenga las mejores características de velocidad-distorsión entre las modalidades probadas. El análisis de velocidad-distorsión determina, en general, una magnitud de distorsión (o error) entre un bloque codificado y un bloque original no codificado que se ha codificado para generar el bloque codificado, así como una velocidad de bits (es decir, un número de bits) usada para generar el bloque codificado. La unidad de intrapredicción 48 puede calcular proporciones a partir de las distorsiones y velocidades para los diversos bloques codificados, para determinar qué modalidad de intrapredicción presenta el mejor valor de velocidad-distorsión para el bloque.

20 **[0141]** En cualquier caso, tras seleccionar una modalidad de intrapredicción para un bloque, la unidad de intrapredicción 48 puede proporcionar información, que indica la modalidad de intrapredicción seleccionada para el bloque, a la unidad de codificación por entropía 56. La unidad de codificación por entropía 56 puede codificar la información que indique la modalidad de intrapredicción seleccionada de acuerdo a las técnicas de esta divulgación. El codificador de vídeo 20 puede incluir datos de configuración en el flujo de bits transmitido, que pueden incluir una pluralidad de tablas de índices de modalidades de intrapredicción y una pluralidad de tablas de índices de modalidades de intrapredicción modificadas (también denominadas tablas de correlación de palabras de código), definiciones de contextos de codificación para diversos bloques e indicaciones de una modalidad de intrapredicción más probable, una tabla de índices de modalidades de intrapredicción y una tabla de índices de modalidades de intrapredicción modificadas, a usar para cada uno de los contextos.

35 **[0142]** Después de que la unidad de procesamiento de predicción 42 genera el bloque predictivo para el bloque de vídeo actual, ya sea mediante la interpredicción o la 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 residual en el bloque residual se pueden aplicar a la unidad de procesamiento de transformaciones 52. La unidad de procesamiento de transformaciones 52 transforma los datos de vídeo residuales en coeficientes de transformación residuales usando una transformación, tal como una transformación de coseno discreta (DCT) o una transformación conceptualmente similar. La unidad de procesamiento de transformación 52 puede convertir los datos de vídeo residuales desde un dominio de píxeles a un dominio de transformación, tal como un dominio de frecuencia.

40 **[0143]** La unidad de procesamiento de transformación 52 puede enviar los coeficientes de transformación resultantes a la unidad de procesamiento de cuantización 54. La unidad de procesamiento de cuantización 54 cuantiza los coeficientes de transformación para reducir adicionalmente la velocidad de bits. El proceso de cuantización puede reducir la profundidad de bits asociada a algunos, o a la totalidad, de los coeficientes. El grado de cuantización se puede modificar ajustando un parámetro de cuantización. En algunos ejemplos, la unidad de procesamiento de cuantización 54 luego puede realizar un barrido de la matriz, incluyendo los coeficientes de transformación cuantizados. De forma alternativa, la unidad de codificación por entropía 56 puede realizar el barrido.

50 **[0144]** Tras la cuantización, la unidad de codificación por entropía 56 realiza la codificación por entropía de los coeficientes de transformación cuantizados. Por ejemplo, la unidad de codificación por entropía 56 puede realizar una codificación de longitud variable adaptativa al contexto (CAVLC), una codificación aritmética binaria adaptativa al contexto (CABAC), una codificación aritmética binaria adaptativa al contexto basada en la sintaxis (SBAC), una codificación por entropía por división de intervalos de probabilidad (PIPE) u otros procedimientos o técnicas de codificación por entropía. Tras la codificación por entropía por la unidad de codificación por entropía 56, el flujo de bits codificado se puede transmitir al decodificador de vídeo 30, o archivar para su posterior transmisión o recuperación por el decodificador de vídeo 30. La unidad de codificación por entropía 56 también puede codificar por entropía los vectores de movimiento y los otros elementos sintácticos para el fragmento de vídeo actual que se está codificando.

60 **[0145]** La unidad de procesamiento de cuantización inversa 58 y la unidad de procesamiento de transformación inversa 60 aplican una cuantización inversa y una transformación inversa, respectivamente, para reconstruir el bloque residual en el dominio de píxel, para su posterior uso como un bloque de referencia de una imagen de referencia. La unidad de compensación de movimiento y disparidad 46 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 46 también puede aplicar uno o más filtros de interpolación al bloque residual reconstruido para calcular valores de píxeles fraccionarios, para su uso en la estimación de movimiento. El sumador 62 añade el bloque residual reconstruido al bloque de predicción compensado

en movimiento, generado por la unidad de compensación de movimiento y disparidad 46 para generar un bloque de referencia para su almacenamiento en la memoria de imágenes de referencia 64. La unidad de estimación de movimiento y disparidad 44 y la unidad de compensación de movimiento y disparidad 46 pueden usar el bloque de referencia como un bloque de referencia para inter-predecir un bloque en una posterior imagen o trama de vídeo.

[0146] De esta manera, el codificador de vídeo 20 es un ejemplo de un codificador de vídeo que puede configurarse para implementar una o más técnicas ejemplares descritas en esta divulgación. Por ejemplo, la memoria de datos de vídeo 40 almacena datos de vídeo. Los datos de vídeo pueden incluir un componente de vídeo de textura de una vista dependiente y un componente de vista de profundidad que corresponde al componente de vista de textura, cada uno de los cuales ha de codificar el codificador de vídeo 20 en un proceso de codificación de vídeo conforme a la norma 3D-AVC o la norma 3D-HEVC.

[0147] En las técnicas descritas en esta divulgación, el codificador de vídeo 20 puede incluir uno o más procesadores que están configurados para codificar, en un proceso de codificación de vídeo conforme a 3D-AVC o a 3D-HEVC, un componente de vista de textura de una vista dependiente de los datos de vídeo. Como se ha descrito anteriormente, cada vista en una norma 3D-AVC incluye un componente de vista de textura y un componente de vista de profundidad. Existe una vista básica y una o más vistas de realce o dependientes en la norma 3D-AVC, donde los componentes de vista de textura de las una o más vistas de realce o dependientes se pueden predecir entre vistas.

[0148] Para codificar el componente de vista de textura, el codificador de vídeo 20 se puede configurar para evaluar la información de movimiento de uno o más bloques vecinos de un bloque actual en el componente de vista de textura, para determinar si al menos un bloque vecino se predice entre vistas con un vector de movimiento de disparidad que se refiere a una imagen de referencia entre vistas en una vista distinta a la vista dependiente. El codificador de vídeo 20 puede obtener un vector de disparidad para el bloque actual basándose en el vector de movimiento de disparidad para uno de los bloques vecinos. Para la codificación de textura en primer lugar, el codificador de vídeo 20 puede codificar un componente de vista de profundidad, de los datos de vídeo, que corresponde al componente de vista de textura posterior a la codificación del componente de vista de textura.

[0149] En algunos ejemplos, la unidad de procesamiento de predicción 42 del codificador de vídeo 20 puede ser un ejemplo de un procesador configurado para implementar los ejemplos descritos en esta divulgación. En algunos ejemplos, una unidad (por ejemplo, uno o más procesadores) distinta a la unidad de procesamiento de predicción 42 puede implementar los ejemplos descritos anteriormente. En algunos ejemplos, la unidad de procesamiento de predicción 42, en conjunto con otras una o más unidades del codificador de vídeo 20, puede implementar los ejemplos descritos anteriormente. En algunos ejemplos, un procesador del codificador de vídeo 20 (no mostrado en la figura 14) puede, solo o en conjunto con otros procesadores del codificador de vídeo 20, implementar los ejemplos descritos anteriormente.

[0150] La figura 15 es un diagrama de bloques que ilustra un ejemplo de un decodificador de vídeo que puede implementar las técnicas descritas en esta divulgación. Por ejemplo, la figura 15. ilustra el decodificador de vídeo 30 que puede representar un decodificador de vídeo conforme a la norma 3D-AVC o a la norma 3D-HEVC. El decodificador de vídeo 30 se describirá usando cierta terminología de la HEVC, tal como PU, TU y CU, pero debería entenderse que las técnicas descritas con referencia al decodificador de vídeo 30 también pueden realizarse con vídeo codificado de acuerdo a la norma H.264.

[0151] El decodificador de vídeo 30 puede realizar la decodificación de interpredicción o la decodificación de intrapredicción. La figura 15 ilustra el decodificador de vídeo 30. En el ejemplo de la figura 15, el decodificador de vídeo 30 incluye una memoria de datos de vídeo 69, una unidad de decodificación por entropía 70, una unidad de procesamiento de predicción 71, una unidad de procesamiento de cuantización inversa 76, una unidad de procesamiento de transformación inversa 78, un sumador 80 y una memoria de imágenes de referencia 82. La unidad de procesamiento de predicción 71 incluye la unidad de compensación de movimiento y disparidad 72 y la unidad de intrapredicción 74. En algunos ejemplos, el decodificador de vídeo 30 puede realizar una pasada de decodificación generalmente recíproca a la pasada de codificación descrita con respecto al codificador de vídeo 20 de la figura 14.

[0152] La memoria de datos de vídeo 69 puede almacenar datos de vídeo, tales como un flujo de bits de vídeo codificado, a decodificar por parte de los componentes del decodificador de vídeo 30. Los datos de vídeo almacenados en la memoria de datos de vídeo 69 se pueden obtener, por ejemplo, a partir de un dispositivo de almacenamiento 34, a partir de un origen de vídeo local, tal como una cámara, mediante comunicación de red cableada o inalámbrica de datos de vídeo o accediendo a medios físicos de almacenamiento de datos. La memoria de datos de vídeo 69 puede formar un almacén temporal de imágenes codificadas (CPB) que almacene datos de vídeo codificados a partir de un flujo de bits de vídeo codificado.

[0153] La memoria de imágenes de referencia 82 es un ejemplo de una memoria intermedia de imágenes decodificadas (DPB) que almacena datos de vídeo de referencia para su uso en la decodificación de datos de vídeo por el decodificador de vídeo 30 (por ejemplo, en las modalidades de intracodificación o inter-codificación). La memoria de datos de vídeo 69 y la memoria de imágenes de referencia 82 pueden ser formados por cualquiera entre varios dispositivos de memoria, tales como memoria dinámica de acceso aleatorio (DRAM), incluyendo DRAM síncrona

(SDRAM), RAM magnetorresistiva (MRAM), RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. La memoria de datos de vídeo 69 y la memoria de imágenes de referencia 82 pueden ser proporcionadas por el mismo dispositivo de memoria o por dispositivos de memoria independientes. En diversos ejemplos, la memoria de datos de vídeo 69 puede estar en un chip con otros componentes del decodificador de vídeo 30, o fuera de chip con respecto a esos componentes.

[0154] Durante el proceso de decodificación, el decodificador de vídeo 30 recibe un flujo de bits de vídeo codificado que representa bloques de vídeo de un fragmento de vídeo codificado y elementos sintácticos asociados, desde el codificador de vídeo 20. La unidad de decodificación por entropía 70 del decodificador de vídeo 30 decodifica por entropía el flujo de bits para generar coeficientes cuantizados, vectores de movimiento y otros elementos sintácticos. La unidad de decodificación por entropía 70 remite los vectores de movimiento y otros elementos sintácticos a la unidad de procesamiento de predicción 71. El decodificador de vídeo 30 puede recibir los elementos sintácticos en el nivel del fragmento de vídeo y/o el nivel del bloque de vídeo.

[0155] Cuando el fragmento de vídeo se codifica como un fragmento intracodificado (I), la unidad de intrapredicción 74 de la unidad de procesamiento de predicción 71 puede generar datos de predicción para un bloque de vídeo del fragmento de vídeo actual, basándose en una modalidad de intrapredicción señalizada y en datos de bloques previamente decodificados de la trama o imagen actual. Cuando la trama de vídeo se codifica como un fragmento intercodificado (es decir, B o P), la unidad de compensación de movimiento y disparidad 72 de la unidad de procesamiento de predicción 71 produce bloques predictivos para un bloque de vídeo del segmento de vídeo actual, basándose en los vectores de movimiento y otros elementos sintácticos recibidos desde la unidad de decodificación por entropía 70. Los bloques predictivos se pueden producir a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El decodificador de vídeo 30 puede construir las listas de imágenes de referencia (RefImgList0 y RefImgList1) usando técnicas de construcción predeterminadas, basándose en imágenes de referencia almacenadas en la memoria de imágenes de referencia 82.

[0156] La unidad de compensación de movimiento y disparidad 72 determina la información de predicción para un bloque de vídeo del fragmento de vídeo actual, analizando sintácticamente los vectores de movimiento y otros elementos sintácticos, y usa la información de predicción para generar los bloques predictivos para el bloque de vídeo actual que está siendo decodificado. Por ejemplo, la unidad de compensación de movimiento y disparidad 72 usa algunos de los elementos sintácticos recibidos para determinar una modalidad de predicción (por ejemplo, intrapredicción o interpredicción), usada para codificar los bloques de vídeo del fragmento de vídeo, un tipo de fragmento de interpredicción (por ejemplo, fragmento B o fragmento P), información de construcción para una o más de las listas de imágenes de referencia 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 decodificar los bloques de vídeo en el fragmento de vídeo actual.

[0157] La unidad de compensación de movimiento y disparidad 72 puede configurarse para realizar las técnicas de ARP descritas en esta divulgación. Como ejemplo, para un bloque actual predicho bidireccionalmente, codificado usando la ARP, la unidad de compensación de movimiento y disparidad 72 puede determinar un primer vector de movimiento de disparidad para el bloque actual y, usando el primer vector de movimiento de disparidad, ubicar un primer bloque correspondiente del bloque actual en una segunda vista. La unidad de compensación de movimiento y disparidad 72 también puede determinar un segundo vector de movimiento de disparidad para el bloque actual y, utilizando el segundo vector de movimiento de disparidad, ubicar un segundo bloque correspondiente del bloque actual en una tercera vista. A partir de la información de movimiento del primer bloque correspondiente y del segundo bloque correspondiente, la unidad de compensación de movimiento y disparidad 72 puede determinar un único vector de movimiento. La unidad de compensación de movimiento y disparidad 72 puede usar este único vector de movimiento para determinar un bloque de referencia del bloque actual, un bloque de referencia del primer bloque correspondiente y un bloque de referencia del segundo bloque correspondiente. La unidad de compensación de movimiento y disparidad 72 puede generar un primer bloque predictivo basándose en el primer bloque correspondiente, el bloque de referencia del bloque actual y el bloque de referencia del primer bloque correspondiente, y generar un segundo bloque predictivo basándose en el segundo bloque correspondiente, el bloque de referencia del bloque actual y el bloque de referencia del segundo bloque correspondiente.

[0158] La unidad de compensación de movimiento y disparidad 72 también puede configurarse, por ejemplo, para que la unidad de compensación de movimiento y disparidad 72 pueda determinar que un bloque actual de una primera vista se codifique utilizando una modalidad de predicción residual avanzada (ARP) y que el bloque actual se prediga de manera bidireccional. Para un bloque de luma del bloque actual, la unidad de compensación de movimiento y disparidad 72 puede realizar la ARP para una primera dirección de predicción, para determinar un primer bloque predictivo del bloque de luma y realizar la ARP para una segunda dirección de predicción, para determinar un segundo bloque predictivo del bloque de luma. Para un bloque de croma del bloque actual, la unidad de compensación de movimiento y disparidad 72 puede realizar la ARP para solo una entre la primera dirección de predicción y la segunda dirección de predicción, para determinar un primer bloque predictivo del bloque de croma.

[0159] La unidad de compensación de movimiento y disparidad 72 también puede configurarse, por ejemplo, para que la unidad de compensación de movimiento y disparidad 72 pueda determinar que un bloque actual de una primera

vista se codifique utilizando una modalidad de ARP. Para un bloque de luma del bloque actual, la unidad de compensación de movimiento y disparidad 72 puede realizar la ARP para determinar un bloque predictivo del bloque de luma. Para un bloque de croma del bloque actual, la unidad de compensación de movimiento y disparidad 72 puede determinar si se realiza la ARP para el bloque de croma, en función del tamaño del bloque de croma. Como ejemplo, la unidad de compensación de movimiento y disparidad 72 puede inhabilitar la ARP en respuesta a que el tamaño del bloque de croma es de 8x8. Como otro ejemplo, la unidad de compensación de movimiento y disparidad 72 puede inhabilitar la ARP en respuesta a que el tamaño del bloque de croma es menor que 32x32. Como otro ejemplo, la unidad de compensación de movimiento y disparidad 72 puede inhabilitar la ARP en respuesta a que el tamaño del bloque de croma es igual a NxN y que el bloque actual comprende una sub-PU, en donde N es igual a uno entre 8, 16, 32 o 64. Como otro ejemplo, la unidad de compensación de movimiento y disparidad 72 puede realizar la ARP en respuesta a que el tamaño del bloque de croma es NxN y que el bloque actual comprende una PU. Como otro ejemplo, la unidad de compensación de movimiento y disparidad 72 puede inhabilitar la ARP en respuesta a que el tamaño del bloque de croma es igual a NxN y que el bloque actual comprende una sub-PU, y realizar la ARP en respuesta a que el tamaño del bloque de croma es MxM y que el bloque actual comprende una PU, en donde N y M son iguales a uno entre 8, 16, 32 y 64, y en donde M es más pequeño que N.

[0160] La unidad de compensación de movimiento y disparidad 72 también puede realizar la interpolación basándose en filtros de interpolación. La unidad de compensación de movimiento y disparidad 72 puede usar filtros de interpolación como los usados por el codificador de vídeo 20 durante la codificación de los bloques de vídeo, para calcular valores interpolados para píxeles fraccionarios de bloques de referencia. En este caso, la unidad de compensación de movimiento y disparidad 72 puede determinar los filtros de interpolación utilizados por el codificador de vídeo 20 a partir de los elementos sintácticos recibidos, y utilizar los filtros de interpolación para generar bloques predictivos.

[0161] La unidad de procesamiento de cuantización inversa 76 realiza la cuantización inversa, es decir, decuantiza, los coeficientes de transformación cuantizados, proporcionados en el flujo de bits y decodificados por la unidad de decodificación por entropía 70. El proceso de cuantización inversa puede incluir el uso de un parámetro de cuantización calculado por el codificador de vídeo 20 para cada bloque de vídeo en el fragmento de vídeo, con el fin de determinar un grado de cuantización y, asimismo, un grado de cuantización inversa que se debería aplicar. La unidad de procesamiento de transformación inversa 78 aplica una transformación inversa (por ejemplo, una DCT inversa, una transformación inversa entera o un proceso de transformación inversa conceptualmente similar) a los coeficientes de transformación, a fin de producir bloques residuales en el dominio de píxeles.

[0162] Después de que la unidad de compensación de movimiento y disparidad 72 genera el bloque predictivo para el bloque de vídeo actual, basándose en los vectores de movimiento y otros elementos sintácticos, el decodificador de vídeo 30 forma un bloque de vídeo decodificado sumando los bloques residuales procedentes de la unidad de procesamiento de transformación inversa 78 a los correspondientes bloques predictivos generados por la unidad de compensación de movimiento y disparidad 72. El sumador 80 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 decodificados a fin de eliminar distorsiones de efecto pixelado. También se pueden usar otros filtros de bucle (en el bucle de codificación o bien después del bucle de codificación) para allanar las transiciones de píxeles o mejorar de otro modo la calidad del vídeo. Los bloques de vídeo decodificados en una imagen dada se almacenan luego en la memoria de imágenes de referencia 82, que almacena imágenes de referencia usadas para la posterior compensación de movimiento. La memoria de imágenes de referencia 82 almacena también vídeo decodificado para su presentación posterior en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la figura 1.

[0163] De esta manera, el decodificador de vídeo 30 es un ejemplo de un decodificador de vídeo que puede configurarse para implementar una o más técnicas ejemplares descritas en esta divulgación. Por ejemplo, la memoria de datos de vídeo 69 almacena datos de vídeo. Los datos de vídeo pueden incluir información a partir de la cual el decodificador de vídeo 30 puede decodificar un componente de vídeo de textura de una vista dependiente y un componente de vista de profundidad que corresponde al componente de vista de textura, cada uno de los cuales codificó el codificador de vídeo 20 en un proceso de codificación de vídeo conforme a la norma 3D-AVC o conforme a la norma 3D-HEVC.

[0164] En las técnicas descritas en esta divulgación, el decodificador de vídeo 30 puede incluir uno o más procesadores que están configurados para decodificar, en un proceso de codificación de vídeo conforme a la norma 3D-AVC o conforme a la norma 3D-HEVC, un componente de vista de textura de una vista dependiente de los datos de vídeo. Para decodificar el componente de vista de textura, el decodificador de vídeo 30 se puede configurar para evaluar la información de movimiento de uno o más bloques vecinos de un bloque actual en el componente de vista de textura, para determinar si al menos un bloque vecino está predicho entre vistas con un vector de movimiento de disparidad que se refiere a una imagen de referencia entre vistas en una vista distinta a la vista dependiente. El decodificador de vídeo 30 puede obtener un vector de disparidad para el bloque actual basándose en el vector de movimiento de disparidad para uno de los bloques vecinos. Para la codificación de textura en primer lugar, el decodificador de vídeo 30 puede decodificar un componente de vista de profundidad, de los datos de vídeo, que corresponde al componente de vista de textura posterior a la decodificación del componente de vista de textura.

[0165] En algunos ejemplos, la unidad de procesamiento de predicción 71 del decodificador de vídeo 30 puede ser un ejemplo de un procesador configurado para implementar los ejemplos descritos en esta divulgación. En algunos ejemplos, una unidad (por ejemplo, uno o más procesadores) distinta a la unidad de procesamiento de predicción 71, puede implementar los ejemplos descritos anteriormente. En algunos ejemplos, la unidad de procesamiento de predicción 71, en conjunto con otras una o más unidades del decodificador de vídeo 30, pueden implementar los ejemplos descritos anteriormente. En algunos otros ejemplos más, un procesador del decodificador de vídeo 30 (no mostrado en la figura 15) puede, solo o en conjunto con otros procesadores del decodificador de vídeo 30, implementar los ejemplos descritos anteriormente.

[0166] La figura 16 muestra un procedimiento ejemplar de predicción de un bloque de vídeo de acuerdo a las técnicas de esta divulgación. Las técnicas de la figura 16 pueden, por ejemplo, realizarse mediante la unidad de compensación de movimiento y disparidad 72 del decodificador de vídeo 30 o mediante la unidad de estimación de movimiento y disparidad 44 o la unidad de compensación de movimiento y disparidad 46 del codificador de vídeo 20. Según las técnicas de la figura 16, un codificador de vídeo puede determinar que un bloque actual de una primera vista se codifique utilizando una modalidad de ARP y que el bloque actual se prediga bidireccionalmente (250). El codificador de vídeo puede determinar un primer vector de movimiento de disparidad y un segundo vector de movimiento de disparidad para el bloque actual (252). El codificador de vídeo puede localizar, con el primer vector de movimiento de disparidad, un primer bloque correspondiente del bloque actual en una segunda vista (254). El codificador de vídeo también puede ubicar, con el segundo vector de movimiento de disparidad, un segundo bloque correspondiente del bloque actual en una tercera vista (256). El codificador de vídeo puede determinar un vector de movimiento a partir de la información de movimiento de al menos uno entre el primer bloque correspondiente del bloque actual y el segundo bloque correspondiente del bloque actual (258). Usando el vector de movimiento, el codificador de vídeo puede identificar un bloque de referencia del bloque actual en la primera vista, un bloque de referencia del primer bloque correspondiente en la segunda vista y un bloque de referencia del segundo bloque correspondiente en la tercera vista (260). En el ejemplo de la figura 17, la segunda vista y la tercera vista pueden ser la misma vista o bien vistas diferentes, pero habitualmente serán diferentes a la primera vista.

[0167] El codificador de vídeo puede generar un primer bloque predictivo basándose en el primer bloque correspondiente, el bloque de referencia del bloque actual y el bloque de referencia del primer bloque correspondiente (262). El codificador de vídeo puede generar un segundo bloque predictivo basándose en el segundo bloque correspondiente, el bloque de referencia del bloque actual y el bloque de referencia del segundo bloque correspondiente (264). El codificador de vídeo puede, por ejemplo, generar el segundo bloque predictivo determinando un predictor residual que corresponde a una diferencia entre el bloque de referencia del bloque actual y el bloque de referencia del segundo bloque correspondiente. El codificador de vídeo puede agregar el predictor residual al segundo bloque correspondiente para generar el bloque predictivo y puede aplicar un factor de ponderación al predictor residual antes de agregarlo al segundo bloque correspondiente.

[0168] El codificador de vídeo puede, por ejemplo, determinar el vector de movimiento a partir de la información de movimiento de al menos uno entre el primer bloque correspondiente del bloque actual y el segundo bloque correspondiente del bloque actual, en respuesta a no estar disponible un vector de movimiento para el primer bloque correspondiente, utilizando un vector de movimiento cero para el vector de movimiento. En otro ejemplo, el codificador de vídeo puede determinar el vector de movimiento a partir de información de movimiento del al menos uno entre el primer bloque correspondiente del bloque actual y el segundo bloque correspondiente del bloque actual, en respuesta a no estar disponible un vector de movimiento para el primer bloque correspondiente del bloque actual, utilizando un vector de movimiento para el segundo bloque correspondiente del bloque actual como el vector de movimiento. En otro ejemplo, el codificador de vídeo puede determinar el vector de movimiento a partir de la información de movimiento del al menos uno entre el primer bloque correspondiente del bloque actual y el segundo bloque correspondiente del bloque actual, en respuesta a no estar disponible un vector de movimiento para el primer bloque correspondiente del bloque actual y a no estar disponible un vector de movimiento para el segundo bloque correspondiente del bloque actual, utilizando un vector de movimiento cero para el vector de movimiento.

[0169] En algunos escenarios de codificación, el codificador de vídeo puede inhabilitar la ARP. Por ejemplo, no estando disponible un vídeo en respuesta a un vector de movimiento para el primer bloque correspondiente de un segundo bloque actual, el codificador de vídeo puede inhabilitar la ARP. En otro ejemplo, en respuesta a no estar disponible un vector de movimiento para el primer bloque correspondiente de un segundo bloque actual y a no estar disponible un vector de movimiento para el segundo bloque correspondiente del segundo bloque actual, el codificador de vídeo puede inhabilitar la ARP para el segundo bloque actual.

[0170] La figura 17 muestra un procedimiento ejemplar para predecir un bloque de vídeo. Las técnicas de la figura 17 pueden, por ejemplo, realizarse mediante la unidad de compensación de movimiento y disparidad 72 del decodificador de vídeo 30 o mediante la unidad de estimación de movimiento y disparidad 44 o la unidad de compensación de movimiento y disparidad 46 del codificador de vídeo 20. Según las técnicas de la figura 17, un codificador de vídeo puede determinar que un bloque actual de una primera vista se codifique utilizando una modalidad de ARP y que el bloque actual se prediga de manera bidireccional (270). Para un bloque de luma del bloque actual, el codificador de vídeo puede realizar la ARP para una primera dirección de predicción, para determinar un primer bloque predictivo del bloque de luma (272). Para el bloque de luma del bloque actual, el codificador de vídeo puede realizar

la ARP para una segunda dirección de predicción, para determinar un segundo bloque predictivo del bloque de luma (274). Para un bloque de croma del bloque actual, el codificador de vídeo puede realizar la ARP para solo una entre la primera dirección de predicción y la segunda dirección de predicción, para determinar un primer bloque predictivo del bloque de croma (276).

5 **[0171]** La figura 18 muestra un procedimiento ejemplar para predecir un bloque de vídeo. Las técnicas de la figura 18 pueden realizarse, por ejemplo, mediante la unidad de compensación de movimiento y disparidad 72 del decodificador de vídeo 30 o mediante la unidad de estimación de movimiento y disparidad 44 o la unidad de compensación de movimiento y disparidad 46 del codificador de vídeo 20. Según las técnicas de la figura 18, el
10 codificador de vídeo puede determinar que un bloque actual de una primera vista se codifique utilizando una modalidad de ARP (280). Para un bloque de luma del bloque actual, el codificador de vídeo puede realizar la ARP para determinar un bloque

predictivo del bloque de luma (282). Para un bloque de croma del bloque actual, el codificador de vídeo puede determinar si se realiza la ARP para el bloque de croma en función del tamaño del bloque de croma.

15 **[0172]** En un ejemplo, el codificador de vídeo puede determinar si se realiza la ARP para el bloque de croma en función del tamaño del bloque de croma, inhabilitando la ARP en respuesta a que el tamaño del bloque de croma sea de 8x8. En otro ejemplo, el codificador de vídeo puede determinar si se realiza la ARP para el bloque de croma en función del tamaño del bloque de croma, inhabilitando la ARP en respuesta a que el tamaño del bloque de croma sea menor que 32x32. En otro ejemplo, el codificador de vídeo puede determinar si se realiza la ARP para el bloque de croma en función del tamaño del bloque de croma, inhabilitando la ARP en respuesta a que el tamaño del bloque de croma sea igual a NxN y que el bloque actual comprenda una sub-PU, y realizar la ARP en respuesta a que el tamaño del bloque de croma sea NxN y que el bloque actual comprenda una PU. N puede ser igual, por ejemplo, a uno entre 8, 16, 32 o 64. En otro ejemplo, el codificador de vídeo puede determinar si se realiza la ARP para el bloque de croma
20 en función del tamaño del bloque de croma, inhabilitando la ARP en respuesta a que el tamaño del bloque de croma sea igual a NxN y que el bloque actual comprenda una sub-PU, y realizar la ARP en respuesta a que el tamaño del bloque de croma sea MxM y que el bloque actual comprenda una PU. N y M pueden, por ejemplo, ser igual a uno entre 8, 16, 32 y 64, M puede ser más pequeño que N.

30 **[0173]** En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, firmware o cualquier combinación de los mismos. Si se implementan en software, las funciones se pueden almacenar en, o transmitir por, un medio legible por ordenador, como una o más instrucciones o código, y ejecutarse mediante una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que correspondan a un medio tangible tal como medios de almacenamiento de datos o medios de comunicación que incluyan cualquier medio que facilite la transferencia de un programa informático desde un lugar a otro, por ejemplo, de acuerdo a un protocolo de comunicación. De esta manera, los medios legibles por ordenador pueden corresponder en general a (1) medios de almacenamiento tangibles legibles por ordenador que sean no transitorios o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser medios disponibles cualesquiera a los que se pueda acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en esta divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

45 **[0174]** A modo de ejemplo, y no de manera limitativa, tales medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento de disco óptico, almacenamiento de disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que pueda usarse para almacenar código de programa deseado en forma de instrucciones o estructuras de datos y al que pueda accederse mediante un ordenador. Además, cualquier conexión recibe debidamente la denominación de medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde una sede de la Red, un servidor u otro origen remoto usando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas se incluyen en la definición de medio. Sin embargo, debería entenderse que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que, en cambio, se orientan a medios de almacenamiento tangibles no transitorios. El término disco, como se usa en el presente documento, incluye un disco compacto (CD), un disco láser, un disco óptico, un disco versátil digital (DVD), un disco flexible y un disco Blu-ray, donde algunos discos normalmente emiten datos magnéticamente, mientras que otros discos emiten datos ópticamente con láseres. Las combinaciones de lo anterior también deberían incluirse dentro del alcance de los medios legibles por ordenador.

60 **[0175]** Las instrucciones pueden ser ejecutadas por uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), formaciones de compuertas programables in situ (FPGA) u otros circuitos lógicos, integrados o discretos, equivalentes. En consecuencia, el término "procesador", como se usa en el presente documento, se puede referir a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento
65

se puede proporcionar dentro de módulos de hardware y/o software dedicados, configurados para la codificación y la decodificación, o incorporados en un códec combinado. Además, las técnicas se podrían implementar totalmente en uno o más circuitos o elementos lógicos.

- 5 **[0176]** Las técnicas de la presente divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluidos un equipo manual inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). Diversos componentes, módulos o unidades se describen en esta divulgación para enfatizar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no requieren necesariamente su realización mediante diferentes unidades de hardware. En cambio, como se ha descrito
- 10 anteriormente, diversas unidades se pueden combinar en una unidad de hardware de códec, o ser proporcionadas por un grupo de unidades de hardware interoperativas, incluyendo uno o más procesadores, como se ha descrito anteriormente, conjuntamente con software y/o firmware adecuados.

REIVINDICACIONES

1. Un procedimiento de codificación de datos de vídeo, el procedimiento que comprende:

5 determinar que un bloque actual (Actual) de una primera vista se codifica utilizando una modalidad de predicción residual avanzada (ARP), en la que el bloque actual se predice bidireccionalmente;

determinar un primer vector de movimiento de disparidad (DMV0) y un segundo vector de movimiento de disparidad (DMV1) para el bloque actual (Actual);

10 ubicar, con el primer vector de movimiento de disparidad, un primer bloque correspondiente (Base0) para el bloque actual en una segunda vista;

15 ubicar, con el segundo vector de movimiento de disparidad, un segundo bloque correspondiente (Base1) del bloque actual en una tercera vista;

determinar un vector de movimiento (mvbase) a partir de la información de movimiento de al menos uno entre el primer bloque correspondiente (Base0) del bloque actual (Actual) y el segundo bloque correspondiente (Base1) del bloque actual (Actual);

20 identificar un bloque de referencia (ActualRef) del bloque actual en la primera vista, un bloque de referencia (Base0Ref) del primer bloque correspondiente en la segunda vista y un bloque de referencia (Base1Ref) del segundo bloque correspondiente en la tercera vista;

25 generar un primer bloque predictivo basándose en el primer bloque correspondiente (Base0), el bloque de referencia (ActualRef) del bloque actual y el bloque de referencia (Base0Ref) del primer bloque correspondiente;

30 generar un segundo bloque predictivo basándose en el segundo bloque correspondiente (Base1), el bloque de referencia (ActualRef) del bloque actual y el bloque de referencia (Base1Ref) del segundo bloque correspondiente;

predecir el bloque actual (Actual) utilizando el primer bloque predictivo y el segundo bloque predictivo,

35 **caracterizado por que:**

40 el bloque de referencia (ActualRef) del bloque actual (Actual) en la primera vista, el bloque de referencia (Base0Ref) del primer bloque correspondiente (Base0) en la segunda vista y el bloque de referencia (Base1Ref) del segundo bloque correspondiente (Base1) en la tercera vista se identifican utilizando el vector de movimiento determinado (mvbase).

45 2. El procedimiento de la reivindicación 1, en el que generar el segundo bloque predictivo comprende determinar un predictor residual, en donde el predictor residual corresponde a una diferencia entre el bloque de referencia (ActualRef) del bloque actual (Actual) y el bloque de referencia (Base1Ref) del segundo bloque correspondiente (Base1).

3. El procedimiento de la reivindicación 2, en el que generar el segundo bloque predictivo comprende además añadir el predictor residual al segundo bloque correspondiente (Base1).

50 4. El procedimiento de la reivindicación 3, en el que generar el segundo bloque predictivo comprende además aplicar un factor de ponderación al predictor residual.

55 5. El procedimiento de cualquiera de las reivindicaciones 1 a 4, en el que la determinación del vector de movimiento (mvbase) a partir de la información de movimiento del al menos uno entre el primer bloque correspondiente (Base0) del bloque actual (Actual) y el segundo bloque correspondiente (Base1) del bloque actual (Actual) comprende:

60 en respuesta a no estar disponible un vector de movimiento para el primer bloque correspondiente (Base0), usar un vector de movimiento cero para el vector de movimiento (mvbase).

6. El procedimiento de cualquiera de las reivindicaciones 1 a 4, en el que la determinación del vector de movimiento (mvbase) a partir de la información de movimiento del al menos uno entre el primer bloque correspondiente (Base0) del bloque actual (Actual) y el segundo bloque correspondiente (Base1) del bloque actual (Actual) comprende:

65

en respuesta a no estar disponible un vector de movimiento para el primer bloque correspondiente (Base0) del bloque actual (Actual), usar un vector de movimiento para el segundo bloque correspondiente (Base1) del bloque actual (Actual) como el vector de movimiento (mvbase).

- 5 **7.** El procedimiento de cualquiera de las reivindicaciones 1 a 4, en el que la determinación del vector de movimiento (mvbase) a partir de la información de movimiento del al menos uno entre el primer bloque correspondiente (Base0) del bloque actual (Actual) y el segundo bloque correspondiente (Base1) del bloque actual (Actual) comprende:
- 10 en respuesta a no estar disponible un vector de movimiento para el primer bloque correspondiente (Base0) del bloque actual (Actual) y a no estar disponible un vector de movimiento para el segundo bloque correspondiente (Base1) del bloque actual (Actual), utilizar un vector de movimiento cero para el vector de movimiento (mvbase).
- 15 **8.** El procedimiento de cualquiera de las reivindicaciones 1 a 4, que comprende además:
- determinar que un segundo bloque actual de la primera vista se codifique utilizando la modalidad de ARP, en la que el segundo bloque actual se predice bidireccionalmente;
- determinar un primer vector de movimiento de disparidad para el segundo bloque actual;
- 20 ubicar, con el primer vector de movimiento de disparidad para el segundo bloque actual, un primer bloque correspondiente del segundo bloque actual en la segunda vista;
- en respuesta a no estar disponible un vector de movimiento para el primer bloque correspondiente del segundo bloque actual, inhabilitar la ARP.
- 25 **9.** El procedimiento de cualquiera de las reivindicaciones 1 a 4, que comprende además:
- determinar que un segundo bloque actual de la primera vista se codifique utilizando la modalidad de ARP, en la que el segundo bloque actual se predice bidireccionalmente;
- 30 determinar un primer vector de movimiento de disparidad para el segundo bloque actual;
- ubicar, con el primer vector de movimiento de disparidad para el segundo bloque actual, un primer bloque correspondiente del segundo bloque actual en la segunda vista;
- 35 determinar un segundo vector de movimiento de disparidad para el segundo bloque actual;
- ubicar, con el segundo vector de movimiento de disparidad para el segundo bloque actual, un segundo bloque correspondiente del segundo bloque actual en la tercera vista;
- 40 en respuesta a no estar disponible un vector de movimiento para el primer bloque correspondiente del segundo bloque actual y a no estar disponible un vector de movimiento para el segundo bloque correspondiente del segundo bloque actual, inhabilitar la ARP para el segundo bloque actual.
- 45 **10.** Un medio de almacenamiento legible por ordenador que tiene almacenadas en el mismo instrucciones que, cuando se ejecutan, hacen que uno o más procesadores realicen el procedimiento de una cualquiera de las reivindicaciones precedentes.
- 50 **11.** Un dispositivo para codificar vídeo, el dispositivo que comprende un codificador de vídeo configurado para realizar el procedimiento de acuerdo a una cualquiera de las reivindicaciones 1 a 9.

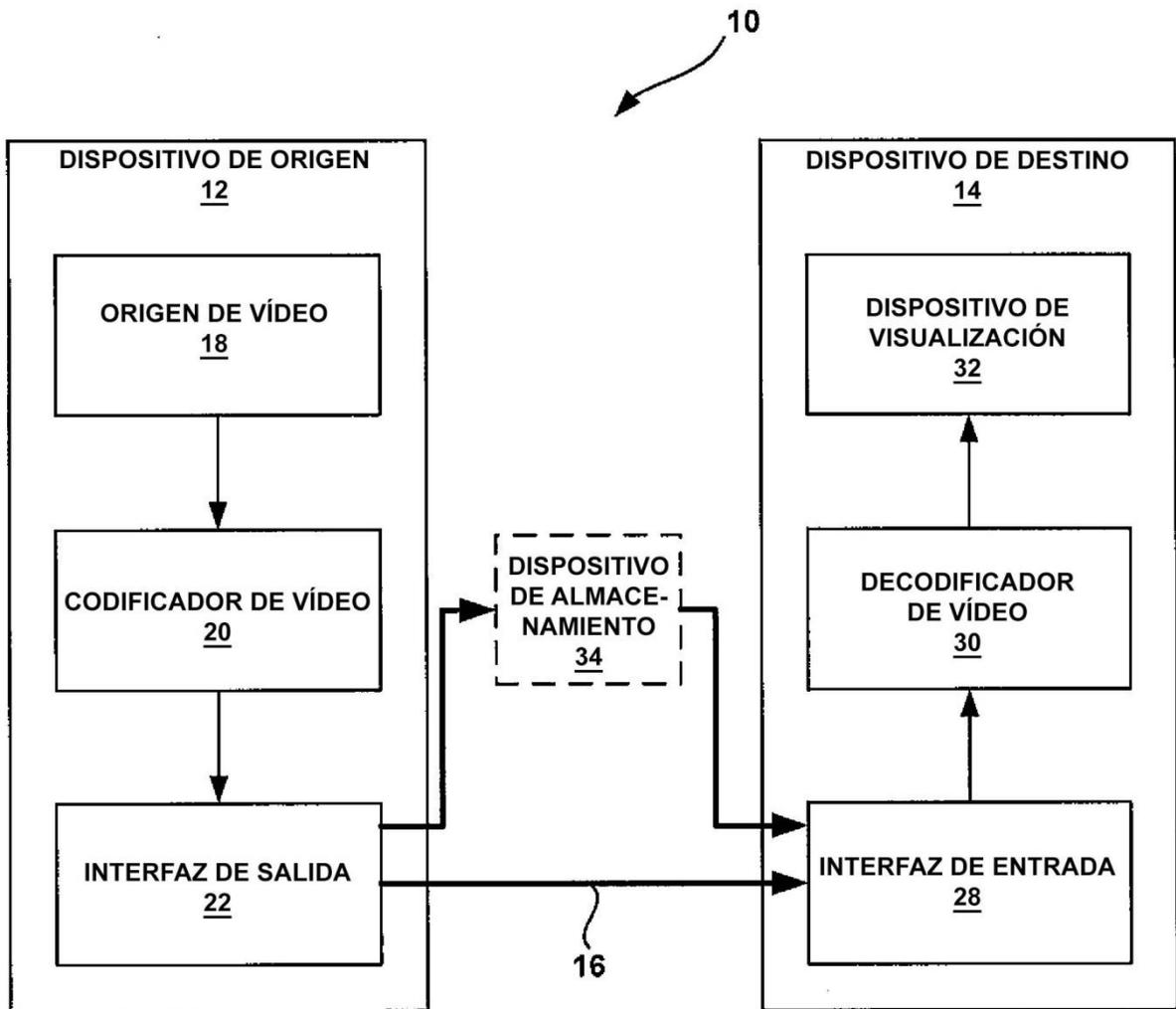


FIG. 1

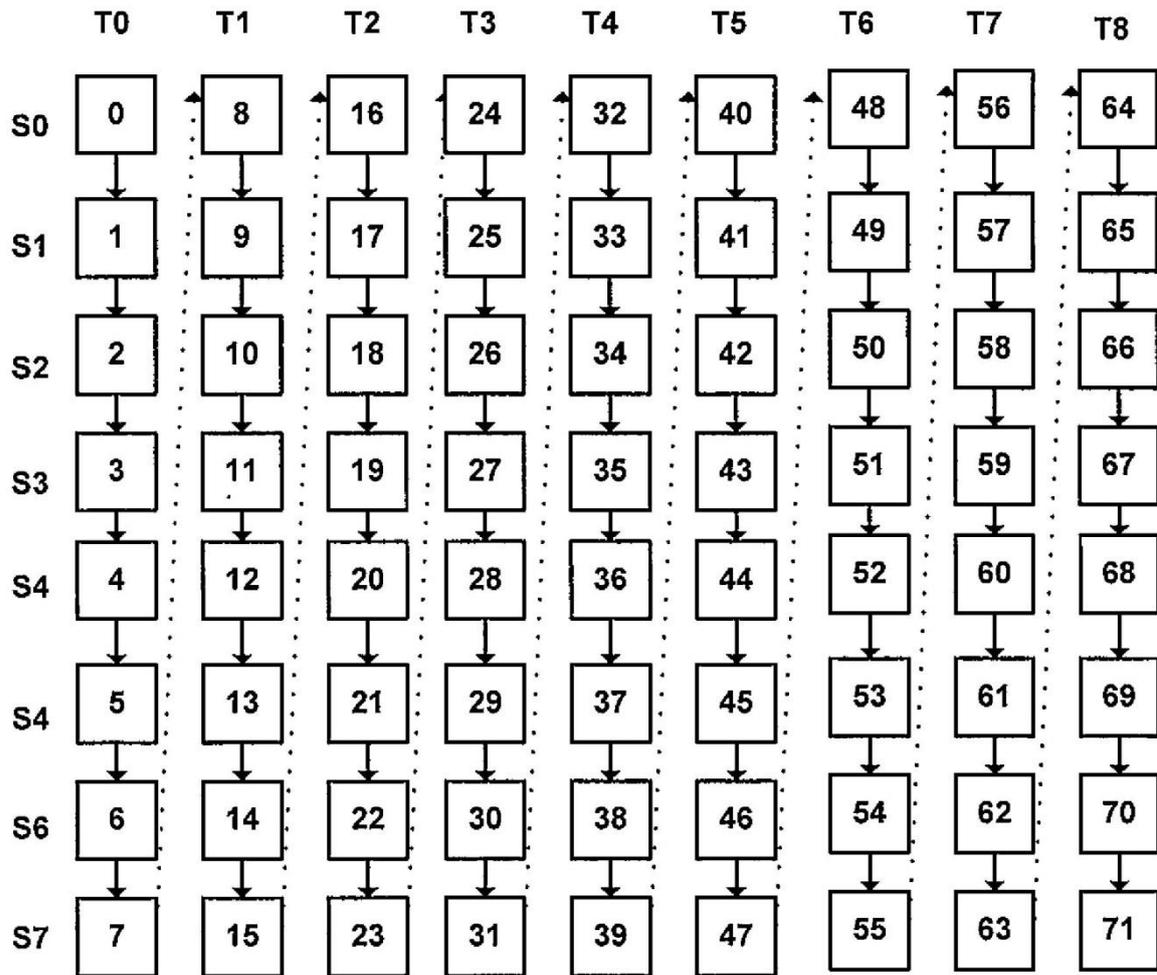


FIG. 2

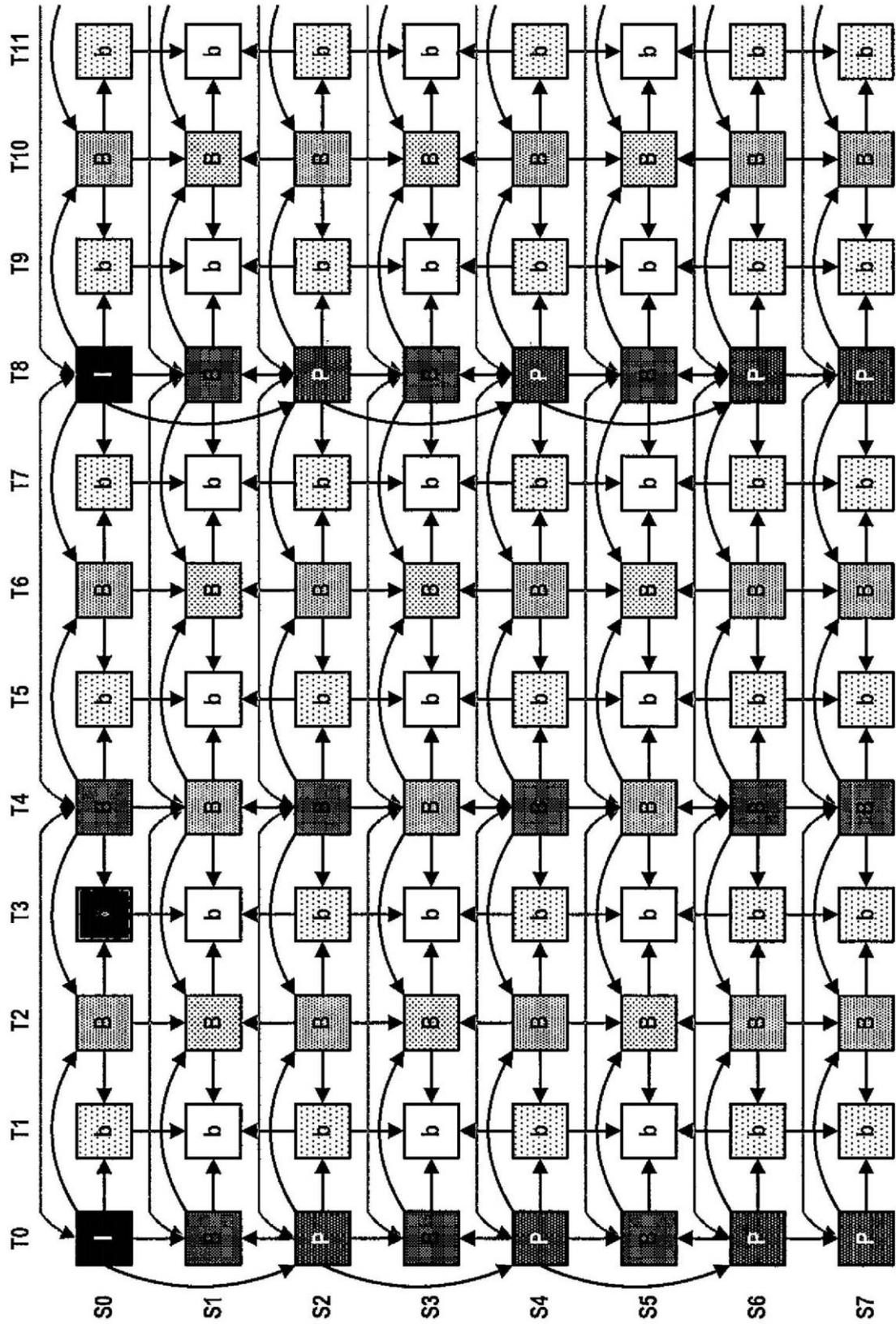


FIG. 3

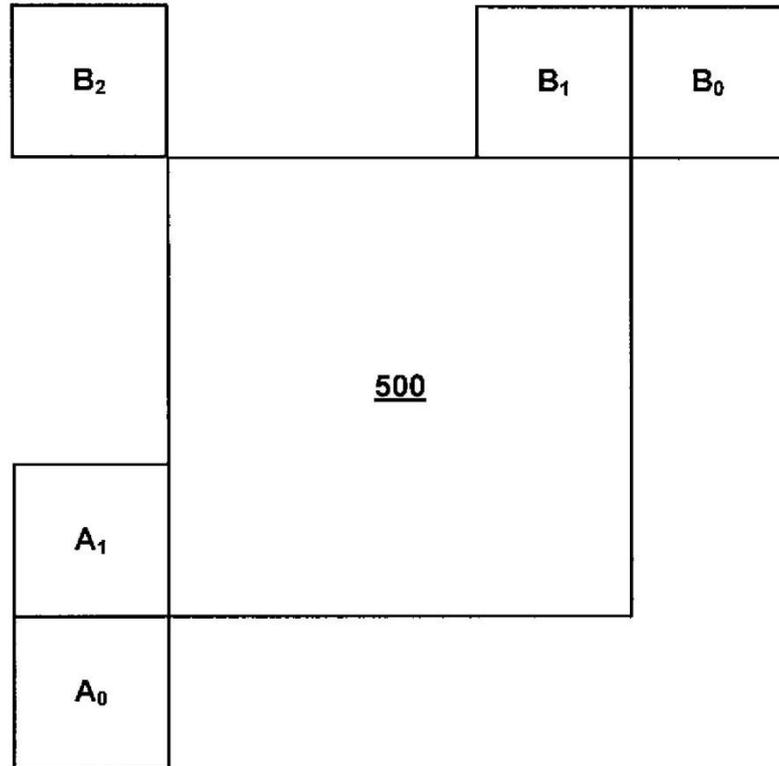


FIG. 4

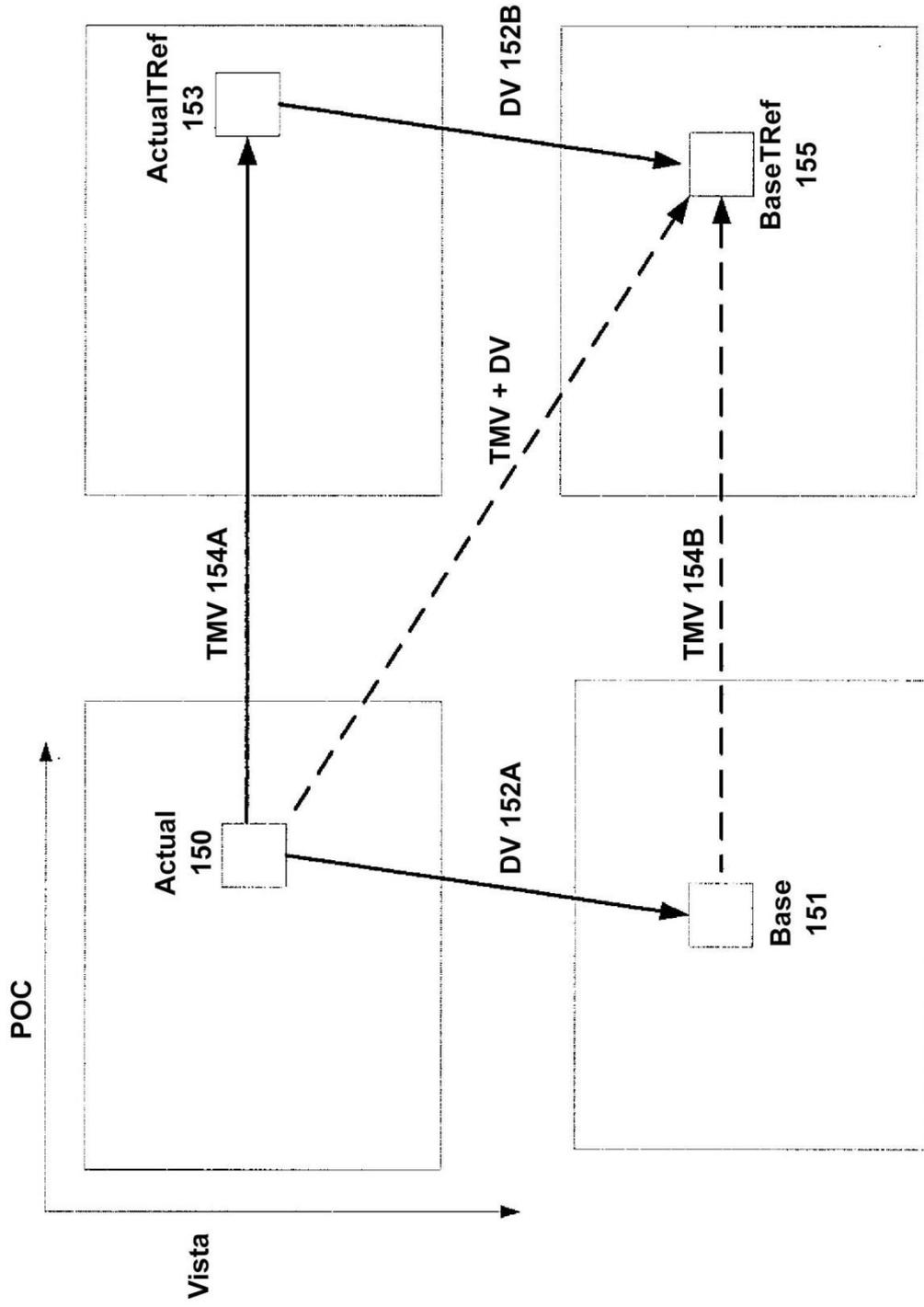


FIG. 5

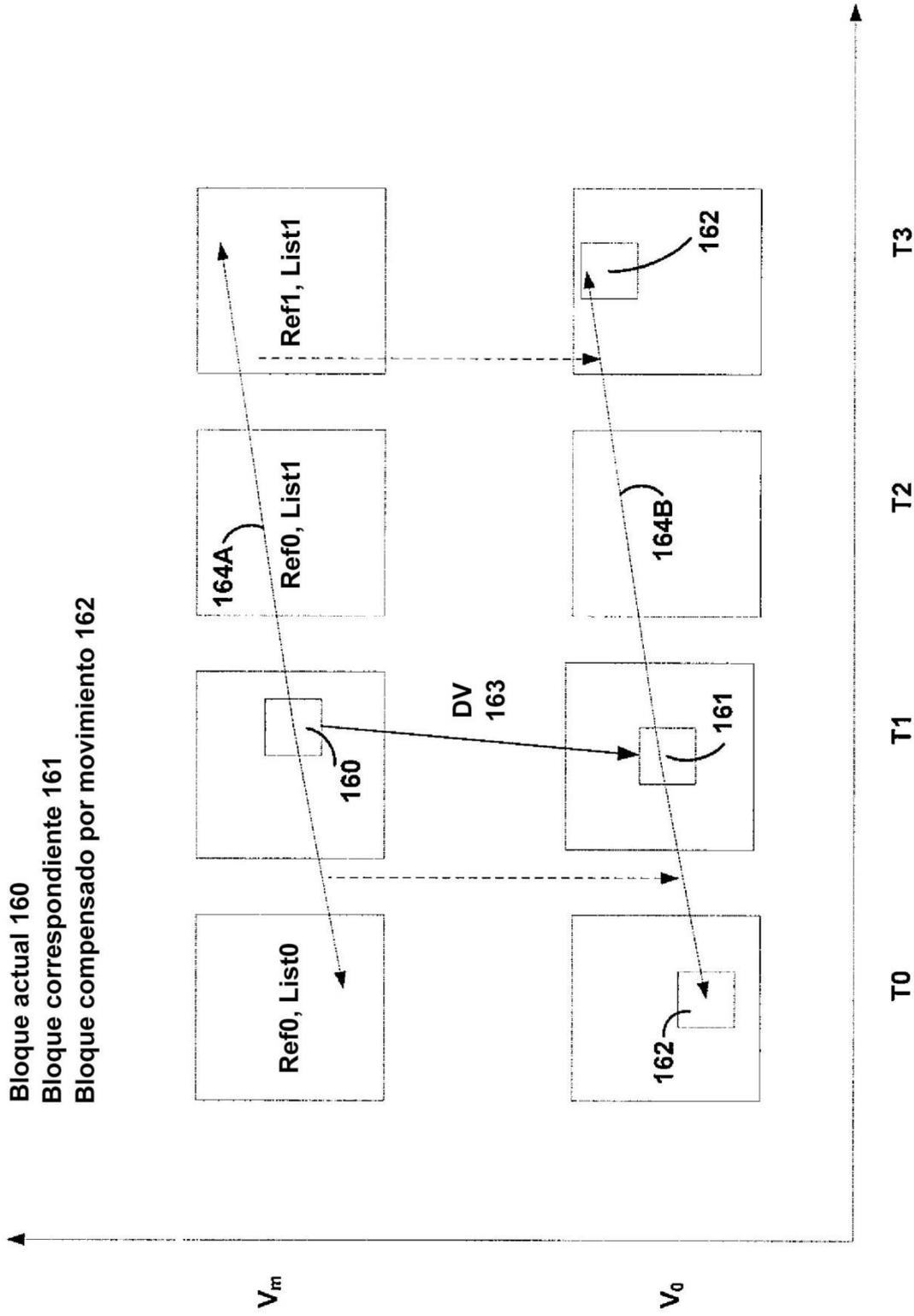


FIG. 6

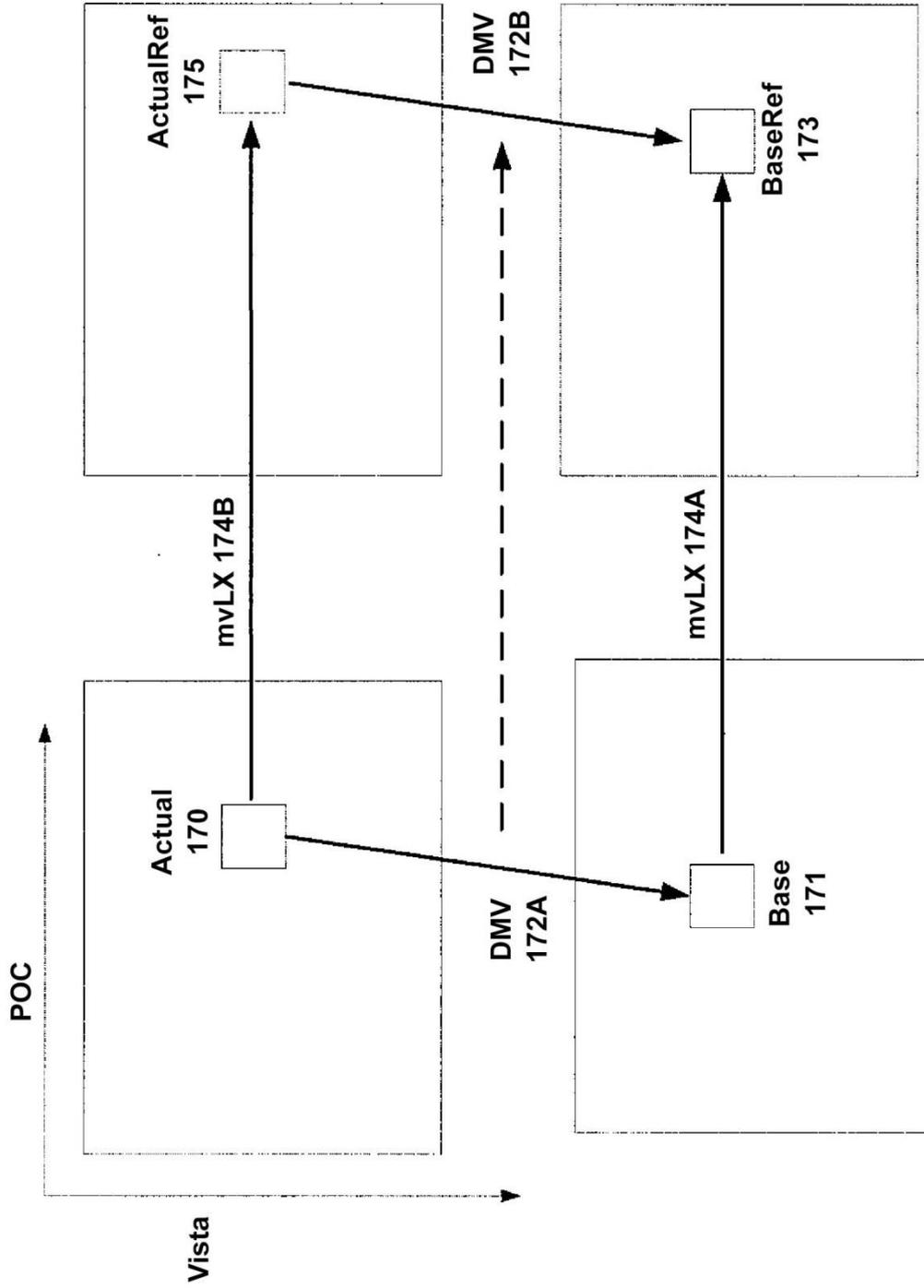


FIG. 7

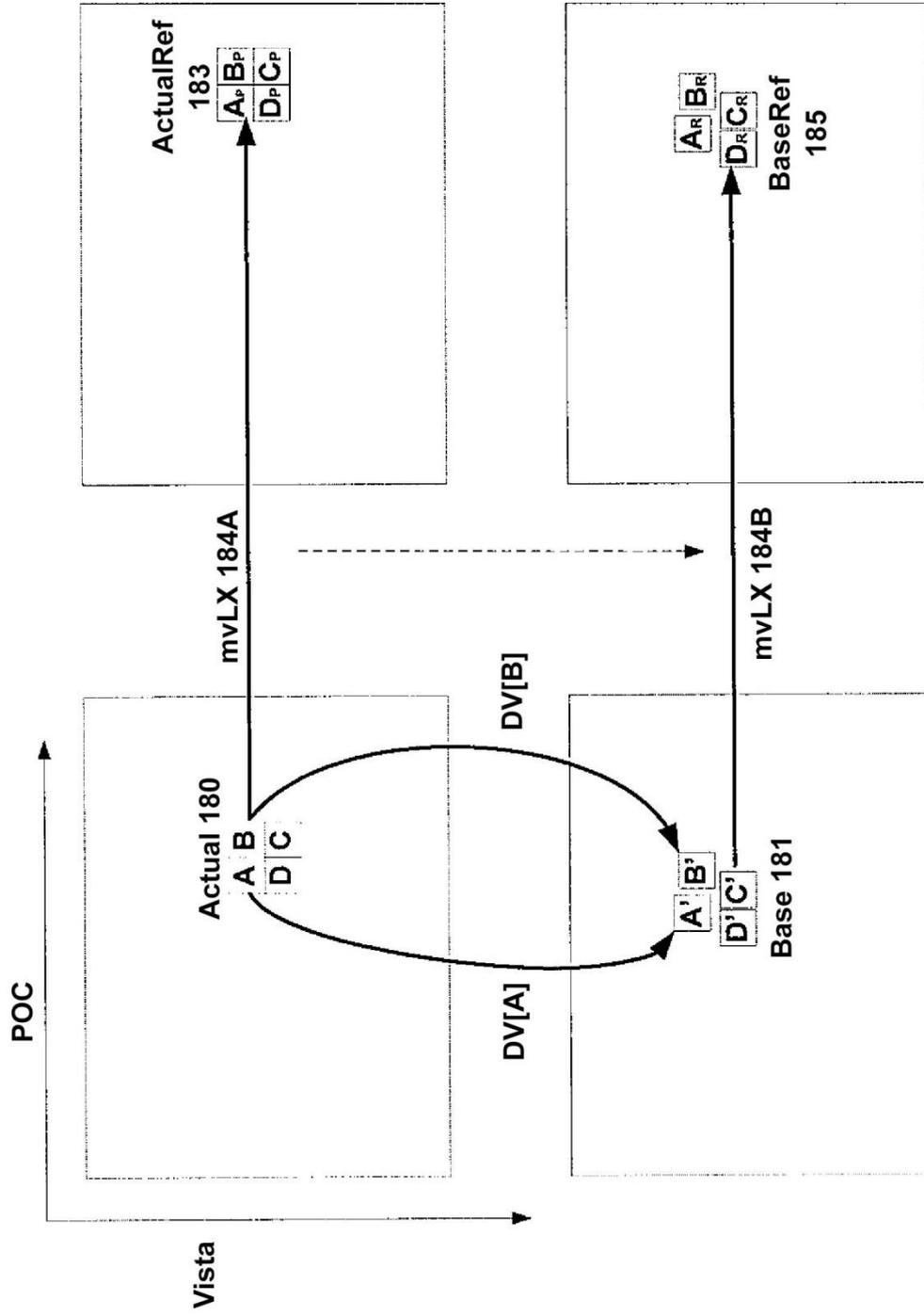


FIG. 8A

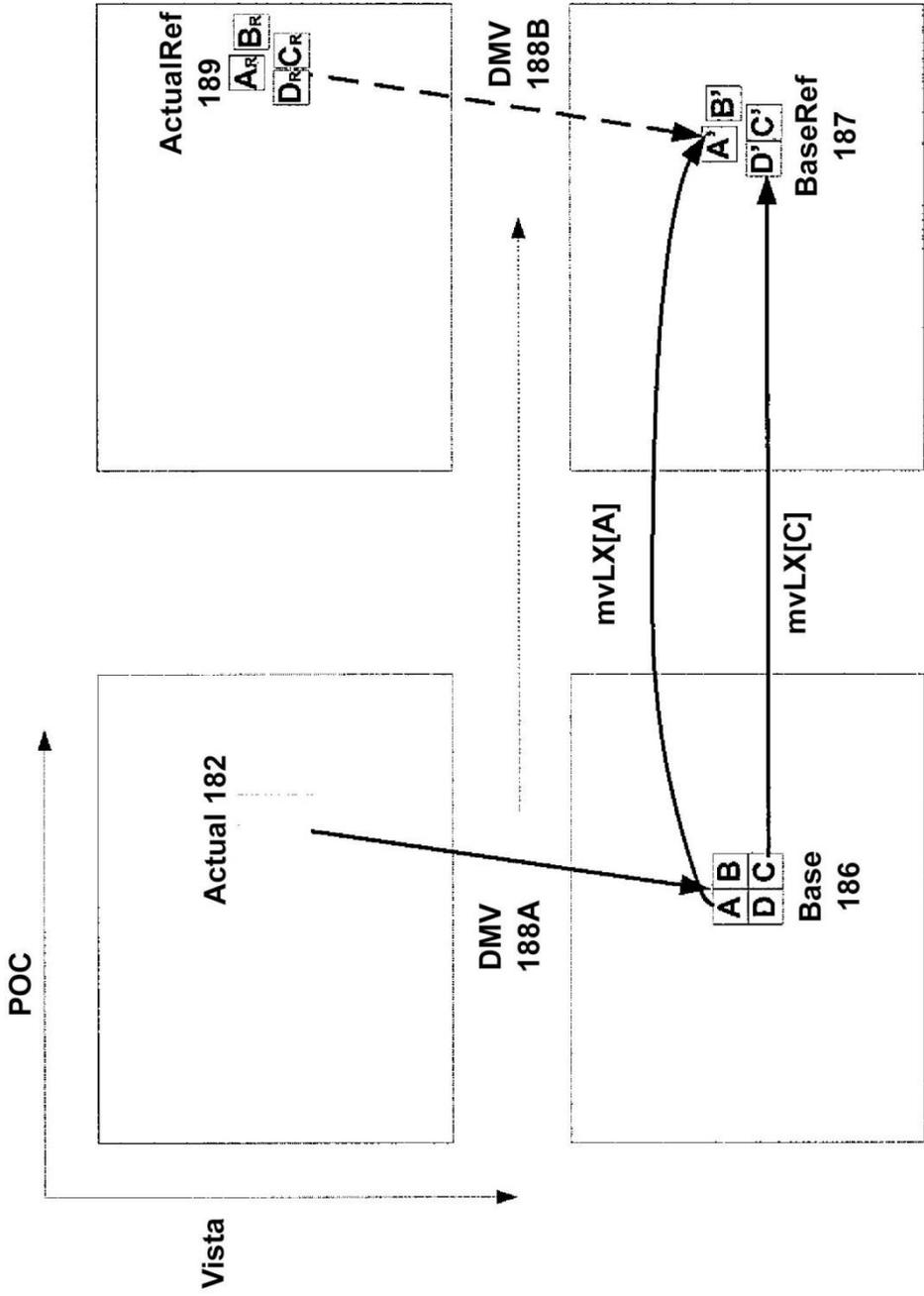


FIG. 8B

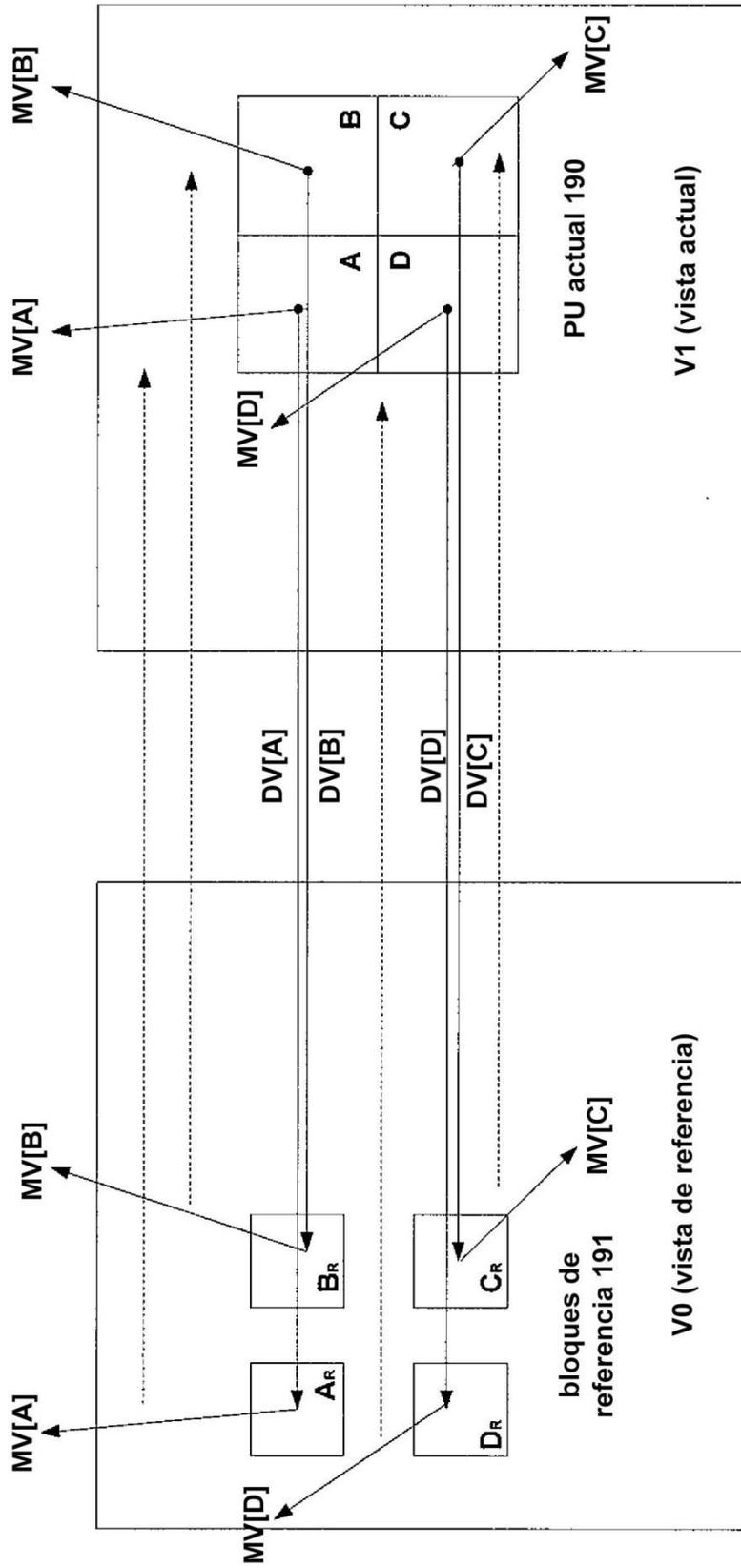


FIG. 9

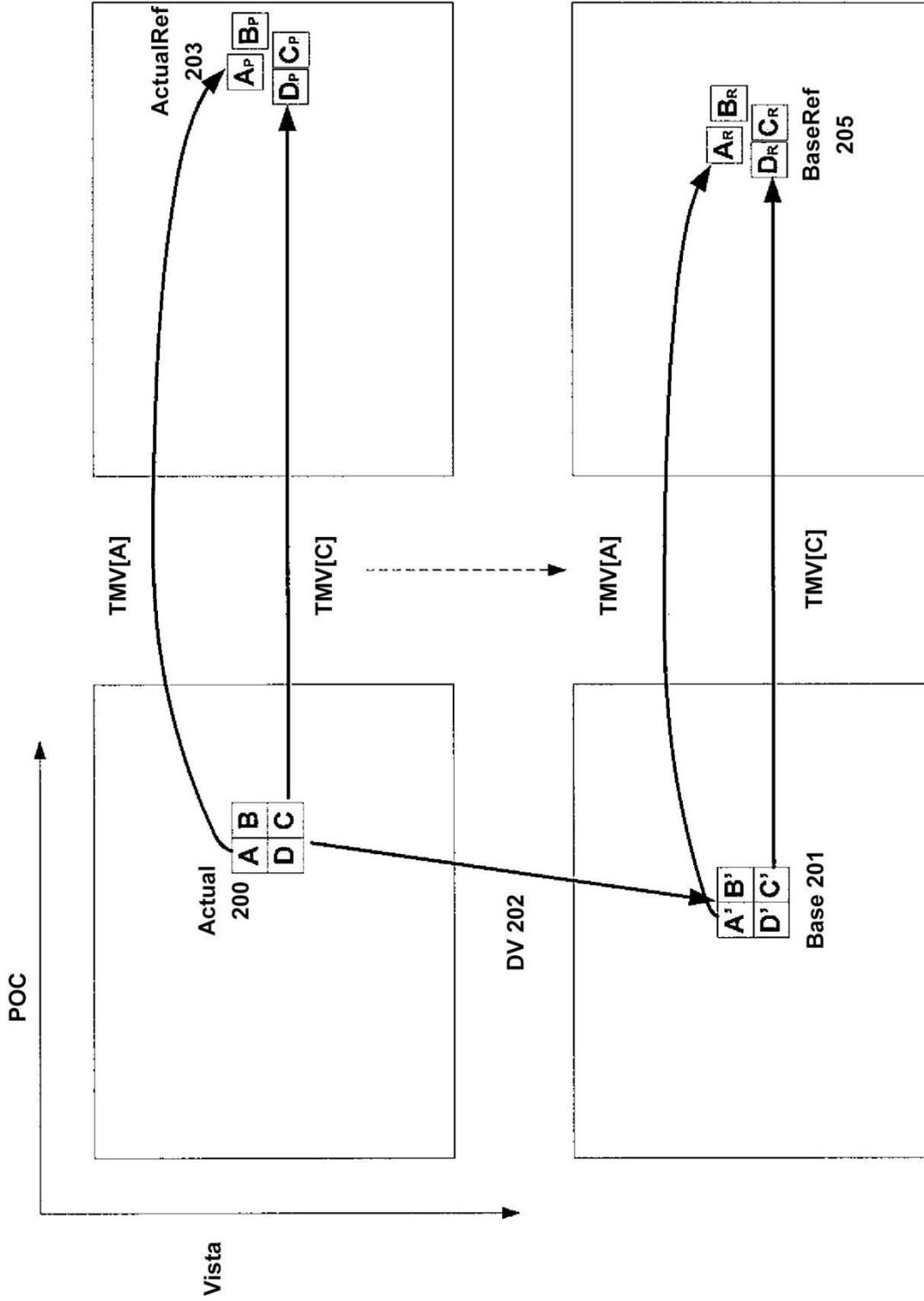


FIG. 10A

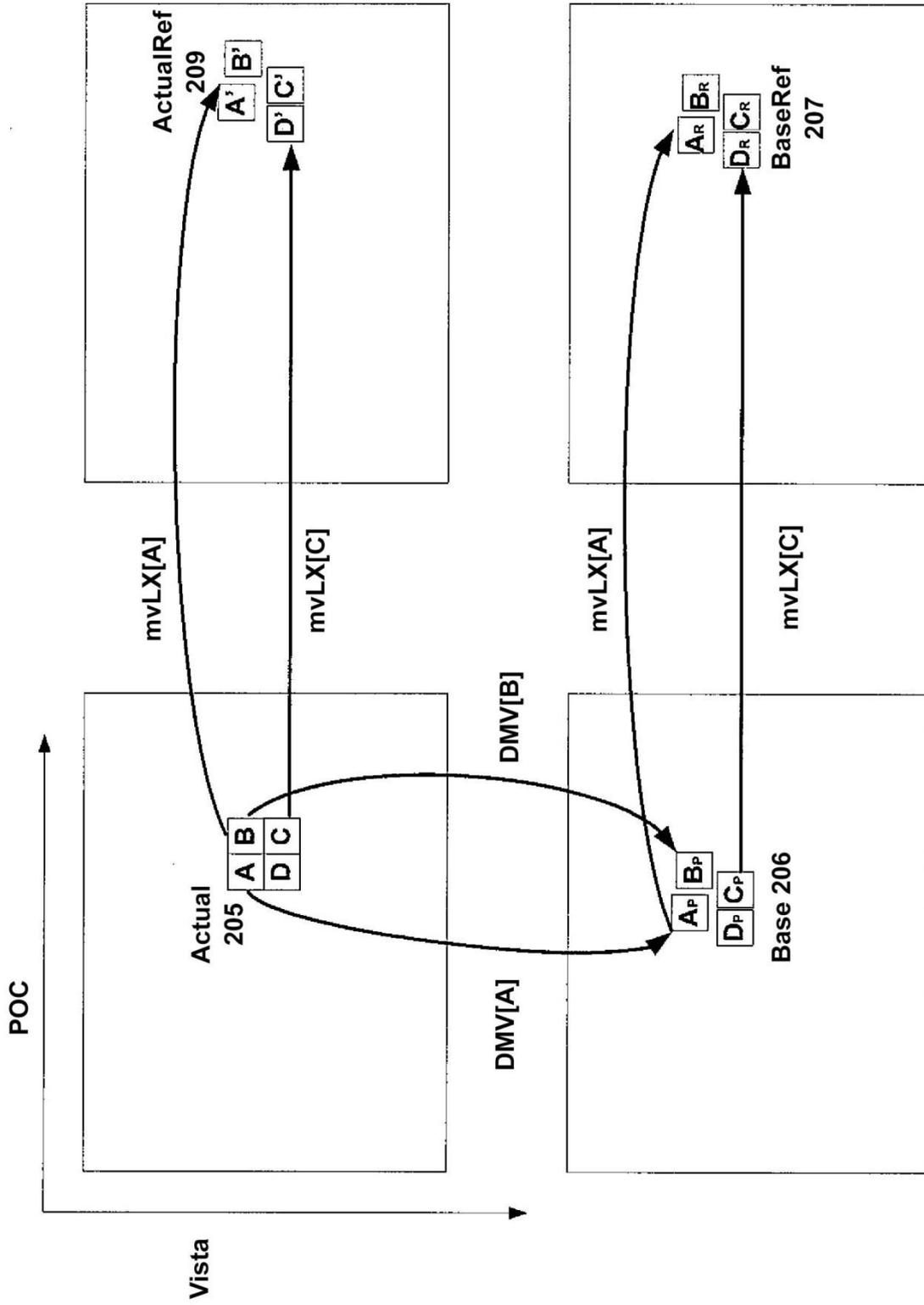
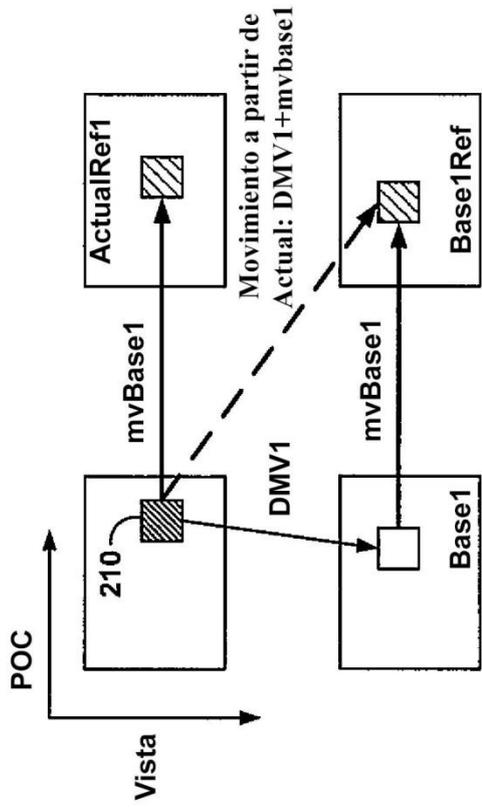
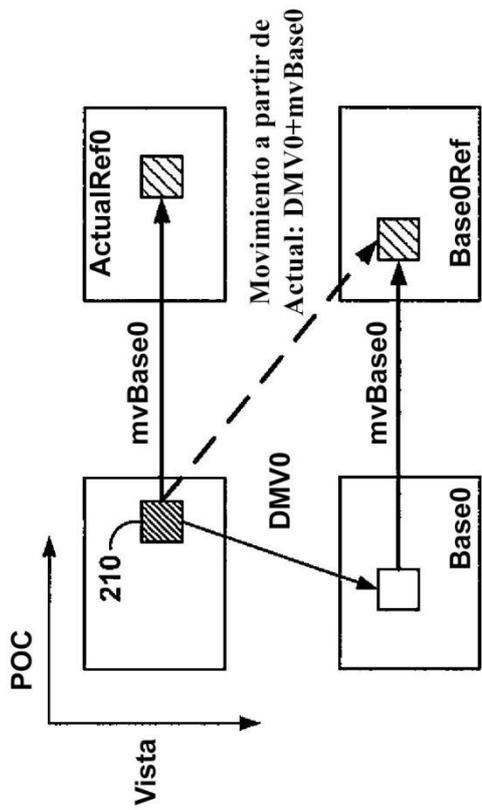


FIG. 10B



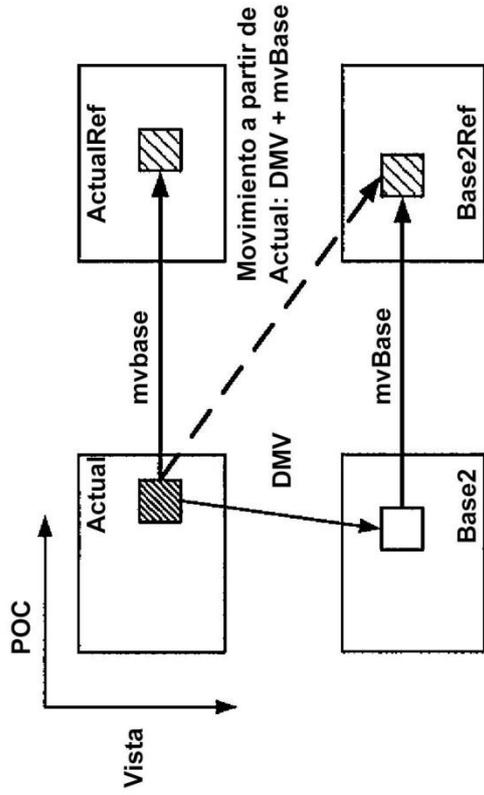
ARP entre vistas para la dirección de predicción 1



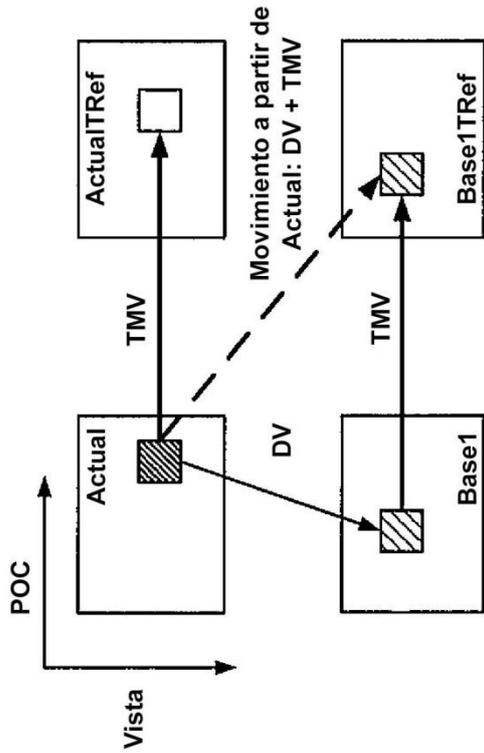
ARP entre vistas para la dirección de predicción 0

-  Bloques de referencia en el proceso de compensación de movimiento
-  Bloques de referencia adicionales utilizados para la ARP
-  Bloque actual 210

FIG. 11



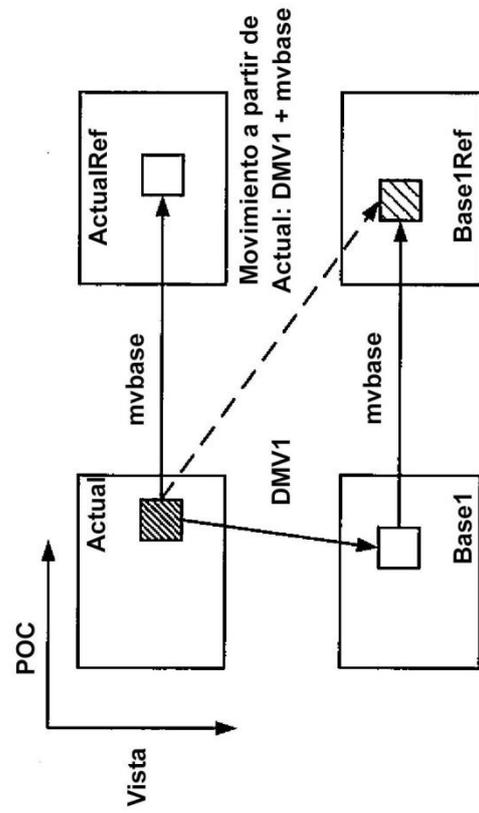
ARP entre vistas



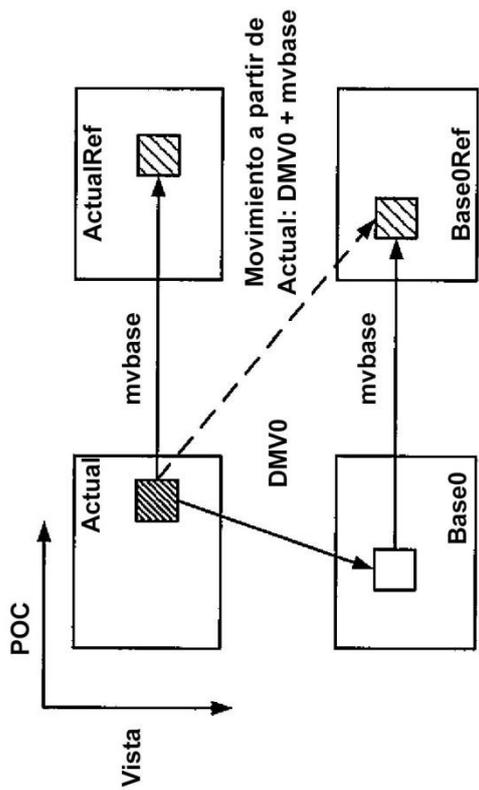
ARP temporal

-  Bloques de referencia en el proceso de compensación de movimiento
-  Bloques de referencia adicionales utilizados para la ARP
-  Bloque actual

FIG. 12



ARP entre vistas para la dirección de predicción 1



ARP entre vistas para la dirección de predicción 0

-  Bloques de referencia en el proceso de compensación de movimiento
-  Bloques de referencia adicionales utilizados para la ARP
-  Bloque actual

FIG. 13

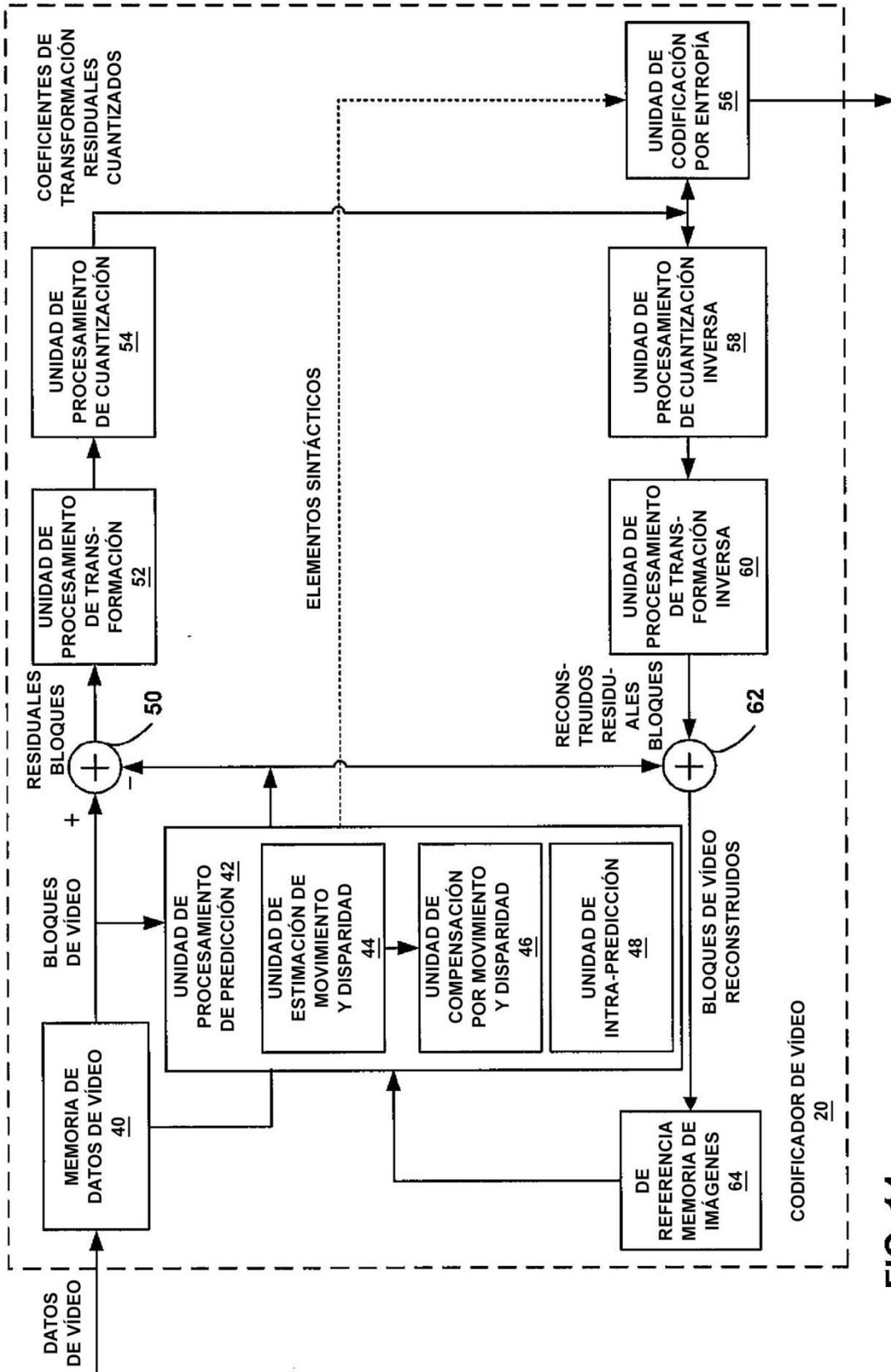


FIG. 14

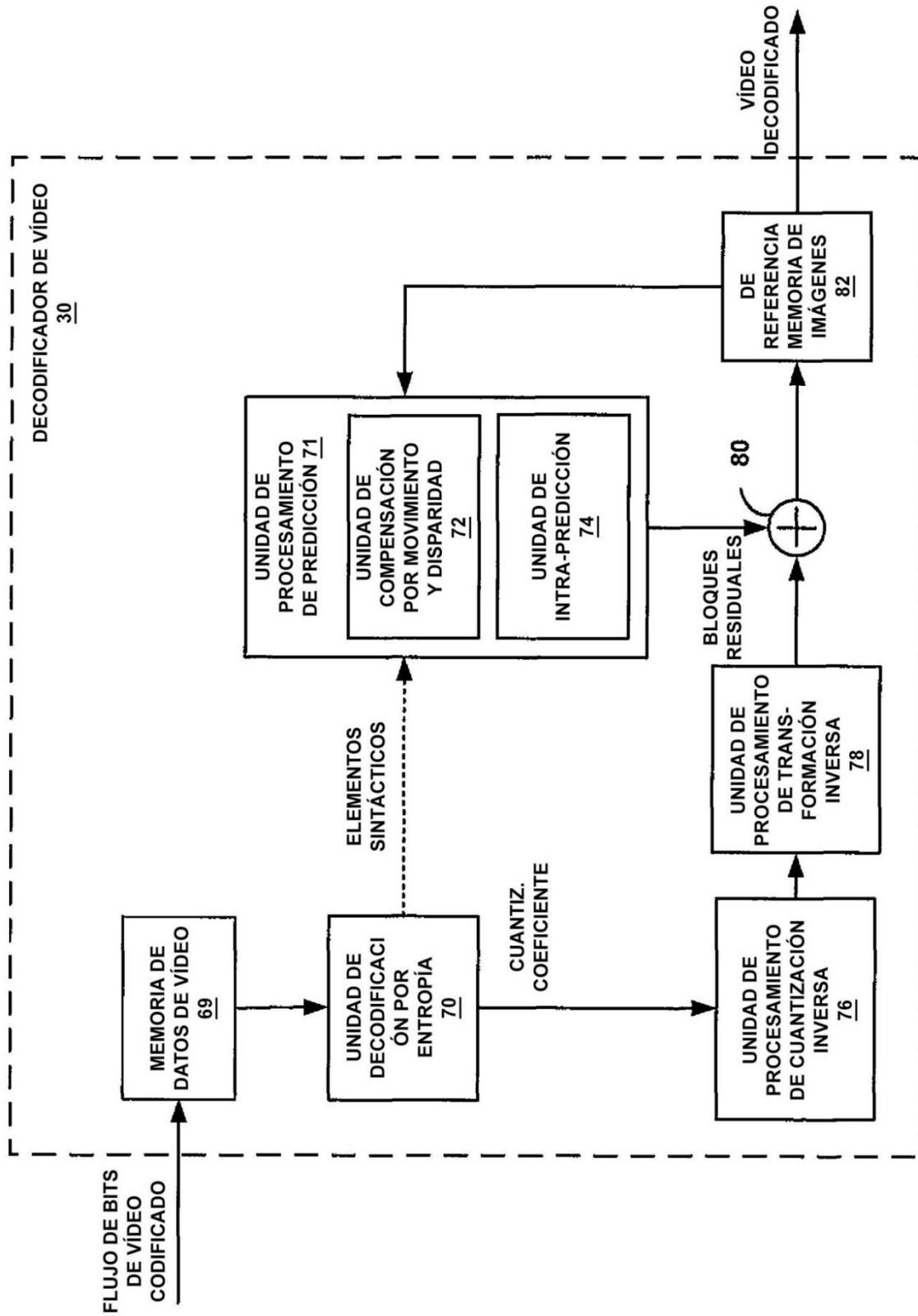


FIG. 15

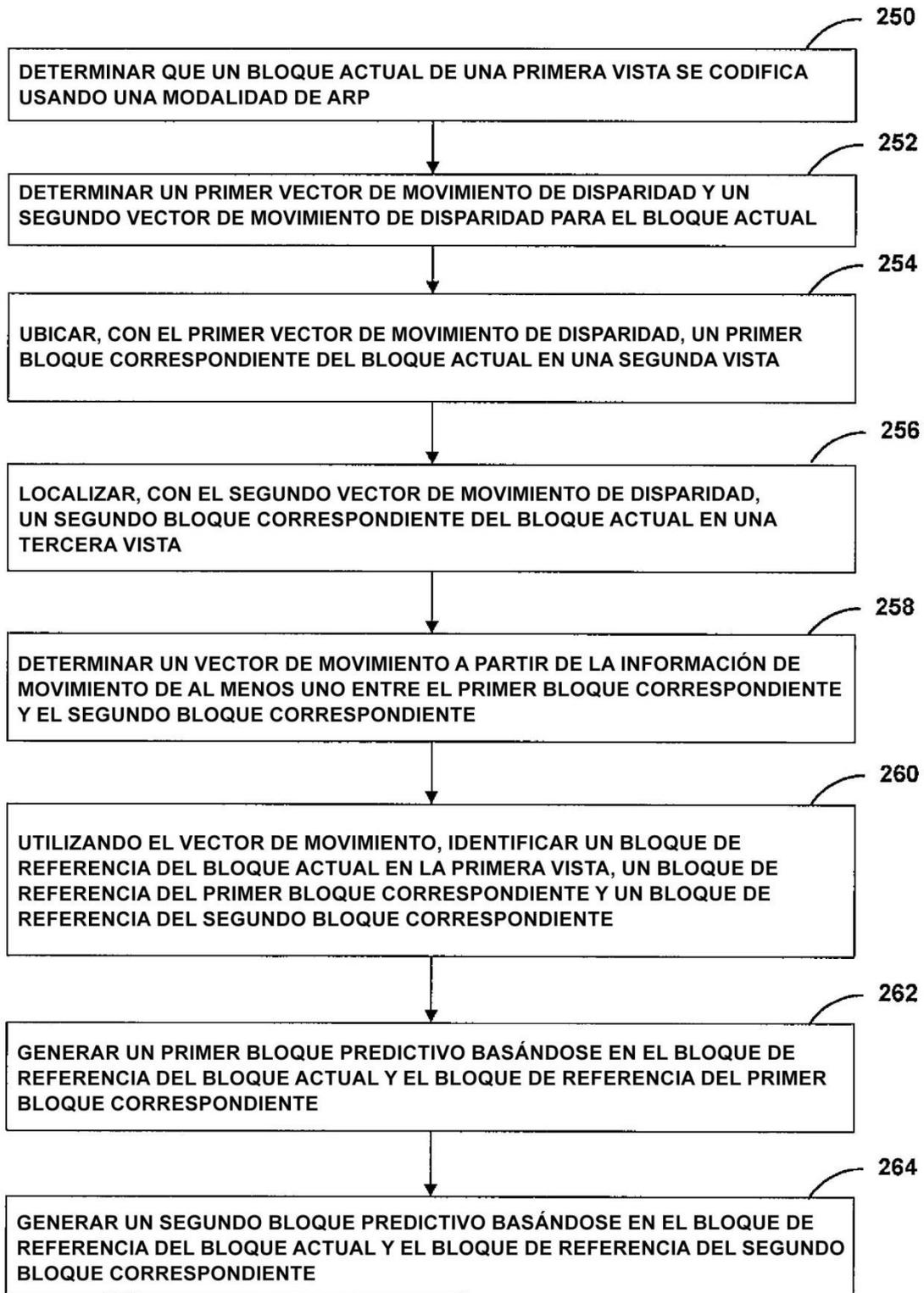


FIG. 16

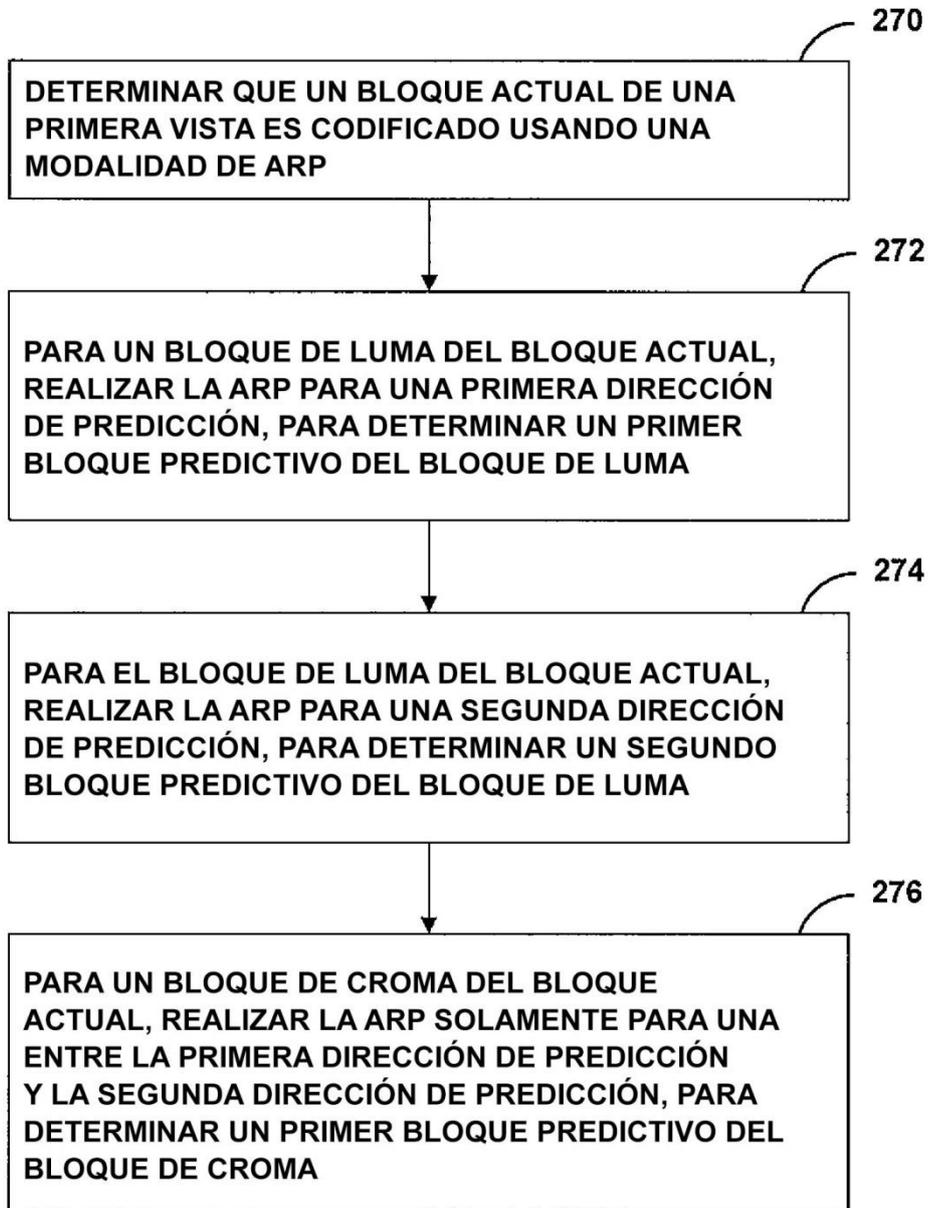


FIG. 17

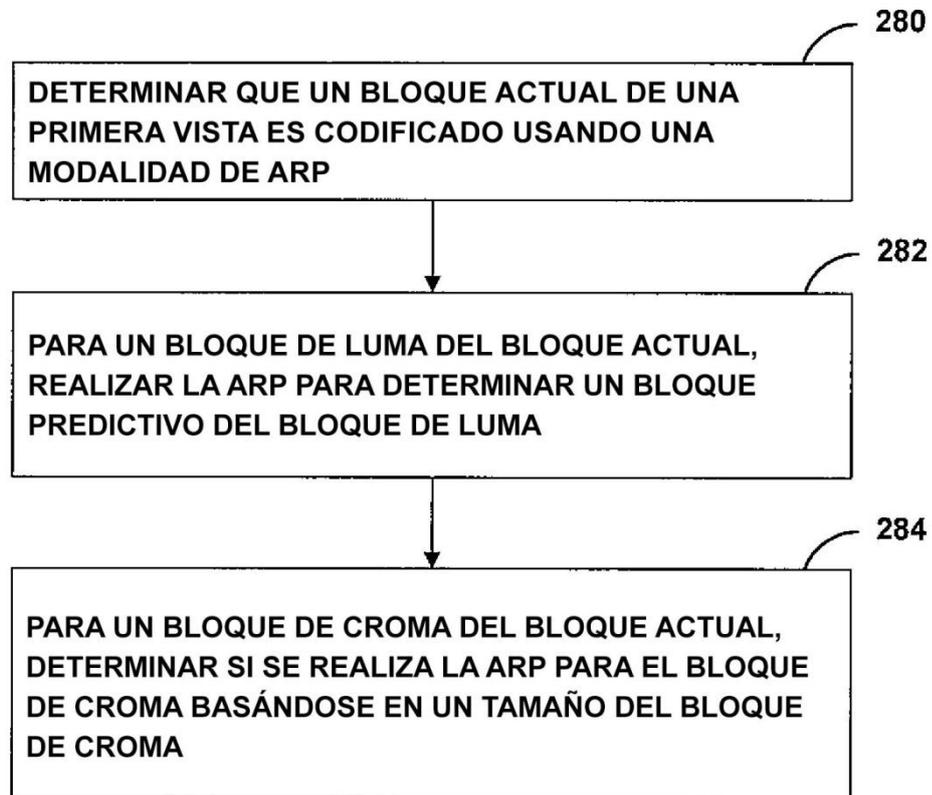


FIG. 18