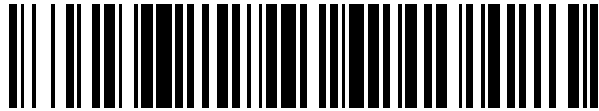


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 799 323**

51 Int. Cl.:

H04N 19/597 (2014.01)
H04N 19/56 (2014.01)
H04N 19/52 (2014.01)
H04N 19/70 (2014.01)
H04N 19/577 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **12.09.2014 PCT/US2014/055456**
- 87 Fecha y número de publicación internacional: **19.03.2015 WO15038937**
- 96 Fecha de presentación y número de la solicitud europea: **12.09.2014 E 14776962 (4)**
- 97 Fecha y número de publicación de la concesión europea: **18.03.2020 EP 3044961**

54 Título: **Técnicas de codificación de vídeo usando particionamiento de movimiento asimétrica**

30 Prioridad:

13.09.2013 US 201361877793 P
23.09.2013 US 201361881383 P
11.09.2014 US 201414483983

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
16.12.2020

73 Titular/es:

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

72 Inventor/es:

CHEN, YING y
ZHANG, LI

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 799 323 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

Técnicas de codificación de vídeo usando particionamiento de movimiento asimétrica

5 **CAMPO TÉCNICO**

[0001] La presente divulgación se refiere a la codificación de vídeo, es decir, codificación o decodificación de datos de vídeo.

10 **ANTECEDENTES**

[0002] Las capacidades de vídeo digital se pueden incorporar en una amplia gama de dispositivos, incluyendo televisores digitales, sistemas de radiodifusión digital directa, sistemas de radiodifusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de sobremesa, ordenadores de tableta, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos móviles o de radio por satélite, los denominados "teléfonos inteligentes", dispositivos de videoconferencia, dispositivos de transmisión continua de vídeo y similares. Los dispositivos de vídeo digital implementan técnicas de codificación de vídeo, tales como las descritas en las normas definidas por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, parte 10, codificación avanzada de vídeo (AVC), la norma de codificación de vídeo de alta eficacia (HEVC), actualmente en fase de desarrollo, y las ampliaciones de dichas normas. Los dispositivos de vídeo pueden transmitir, recibir, codificar, decodificar y/o almacenar información de vídeo digital más eficazmente, implementando dichas técnicas de codificación de vídeo.

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

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

[0005] El documento XP030131250 (SHIMIZU S ET AL.: "3D-CE1,h: Adaptive block partitioning for VSP", 5. CONFERENCIA DEL JCT-3V; 27-7-2013 - 2-8-2013; VIENA; (EL EQUIPO DE COLABORACIÓN CONJUNTA EN DESARROLLO DE EXTENSIONES DE CODIFICACIÓN DE VÍDEO 3D DE ISO/IEC JTC1/SC29/WG11 E ITU-T SG.16); URL: [HTTP://PHENIX.INT-EVRY.FR/JCT2/](http://phenix.int-evry.fr/jct2/), n.º JCT3V-E0207, 19 de julio de 2013 (19-07-2013)); presentación a JCT-3V que propone un tamaño de bloque de 4x8 u 8x4 en el VSP del modelo de prueba 3D-HEVC 4.

[0006] El documento XP030020321 ("3D-HEVC Test Model 4", 104. CONFERENCIA DE MPEG; 22-4-2013 - 26-4-2013; INCHEON (GRUPO DE EXPERTOS DE IMAGEN EN MOVIMIENTO O ISO/IEC JTC1/SC29/WG11), n.º N13573, 24 de junio de 2013 (24-06-2013)); codificador de modelo de prueba 3D-HEVC 4 usando predicción de síntesis de vista (VSP) inversa. El candidato VSP se inserta en la lista de candidatos de fusión.

BREVE EXPLICACIÓN

[0007] En general, la presente divulgación se refiere a la codificación de vídeo tridimensional (3D) en base a códecs avanzados, incluyendo, en algunos ejemplos, técnicas de codificación de profundidad. La presente divulgación describe técnicas para la codificación de predicción de síntesis de vista, incluyendo la determinación de tamaños de bloque, cuando se usa junto con particionamiento de movimiento asimétrica. La presente divulgación también describe técnicas para la predicción de movimiento de avance cuando se usa junto con particionamiento de movimiento asimétrica.

65

[0008] En un ejemplo de la divulgación, un procedimiento de descodificación de datos de vídeo comprende recibir datos residuales correspondientes a un bloque de datos de vídeo, en el que el bloque de datos de vídeo se ha codificado usando particionamiento de movimiento asimétrica, se predice unidireccionalmente usando predicción de síntesis de vista inversa (BVSP) y tiene un tamaño de 16x12, 12x16, 16x4 o 4x16, particionar el bloque de datos de vídeo en subbloques, teniendo cada subbloque un tamaño de 8x4 o 4x8, derivar un respectivo vector de movimiento de disparidad para cada uno de los subbloques de un bloque de profundidad correspondiente en una imagen de profundidad correspondiente a una imagen de referencia, sintetizar un respectivo bloque de referencia para cada uno de los subbloques usando los respectivos vectores de movimiento de disparidad derivados, y descodificar el bloque de datos de vídeo realizando una compensación de movimiento en cada uno de los subbloques usando los datos residuales y los respectivos bloques de referencia sintetizados.

[0009] En otro ejemplo de la divulgación, un procedimiento de codificación de datos de vídeo comprende generar un bloque de datos de vídeo usando particionamiento de movimiento asimétrica, en el que el bloque de datos de vídeo se predice unidireccionalmente usando predicción de síntesis de vista inversa (BVSP) y tiene un tamaño de 16x12, 12x16, 16x4 o 4x16, particionar el bloque de datos de vídeo en subbloques, teniendo cada subbloque un tamaño de 8x4 o 4x8, derivar un respectivo vector de movimiento de disparidad para cada uno de los subbloques de un bloque de profundidad correspondiente en una imagen de profundidad correspondiente a una imagen de referencia, sintetizar un respectivo bloque de referencia para cada uno de los subbloques usando los respectivos vectores de movimiento de disparidad derivados, y codificar el bloque de datos de vídeo realizando una compensación de movimiento en cada uno de los subbloques usando los respectivos bloques de referencia sintetizados.

[0010] En otro ejemplo de la divulgación, un aparato configurado para descodificar datos de vídeo comprende una memoria de vídeo configurada para almacenar información correspondiente a un bloque de datos de vídeo, y uno o más procesadores configurados para recibir datos residuales correspondientes al bloque de datos de vídeo, en el que el bloque de datos de vídeo se ha codificado usando particionamiento de movimiento asimétrica, se predice unidireccionalmente usando predicción de síntesis de vista inversa (BVSP) y tiene un tamaño de 16x12, 12x16, 16x4 o 4x16, particionar el bloque de datos de vídeo en subbloques, teniendo cada subbloque un tamaño de 8x4 o 4x8, derivar un respectivo vector de movimiento de disparidad para cada uno de los subbloques de un bloque de profundidad correspondiente en una imagen de profundidad correspondiente a una imagen de referencia, sintetizar un respectivo bloque de referencia para cada uno de los subbloques usando los respectivos vectores de movimiento de disparidad derivados, y descodificar el bloque de datos de vídeo realizando una compensación de movimiento en cada uno de los subbloques usando los datos residuales y los respectivos bloques de referencia sintetizados.

[0011] En otro ejemplo de la divulgación, un aparato configurado para descodificar datos de vídeo comprende medios para recibir datos residuales correspondientes a un bloque de datos de vídeo, en el que el bloque de datos de vídeo se ha codificado usando particionamiento de movimiento asimétrica, se predice unidireccionalmente usando predicción de síntesis de vista inversa (BVSP) y tiene un tamaño de 16x12, 12x16, 16x4 o 4x16, medios para particionar el bloque de datos de vídeo en subbloques, teniendo cada subbloque un tamaño de 8x4 o 4x8, medios para derivar un respectivo vector de movimiento de disparidad para cada uno de los subbloques de un bloque de profundidad correspondiente en una imagen de profundidad correspondiente a una imagen de referencia, medios para sintetizar un respectivo bloque de referencia para cada uno de los subbloques usando los respectivos vectores de movimiento de disparidad derivados, y medios para descodificar el bloque de datos de vídeo realizando una compensación de movimiento en cada uno de los subbloques usando los datos residuales y los respectivos bloques de referencia sintetizados.

[0012] Los detalles de uno o más ejemplos de la divulgación se exponen en los dibujos adjuntos y la descripción a continuación. La invención se define en las reivindicaciones adjuntas. Los ejemplos, excepto los relacionados con las reivindicaciones, se refieren a ejemplos útiles para entender la invención, pero que no representan modos de realización de la presente invención reivindicada.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

[0013]

La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y descodificación de vídeo de ejemplo que puede utilizar las técnicas de predicción inter de la presente divulgación.

La FIG. 2 es un diagrama conceptual que ilustra un orden de descodificación de ejemplo para vídeo multivista.

La FIG. 3 es un diagrama conceptual que ilustra una estructura de predicción de ejemplo para vídeo multivista.

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

La FIG. 5 es un diagrama conceptual que ilustra tipos de particionamiento de ejemplo.

La FIG. 6 es un diagrama conceptual que ilustra candidatos de vector de movimiento en modo de fusión.

La FIG. 7 es una tabla que indica una especificación de ejemplo de índices de candidatos de fusión.

La FIG. 8 es un diagrama conceptual que ilustra bloques contiguos usados para un procedimiento de derivación de vector de disparidad de ejemplo.

5 La FIG. 9 es un diagrama conceptual que ilustra un procedimiento de derivación de vector de disparidad de bloques contiguos.

10 La FIG. 10 es un diagrama conceptual que ilustra cuatro píxeles de esquina de un bloque de profundidad de 8x8.

La FIG. 11 es un diagrama conceptual que ilustra una derivación de ejemplo del candidato de vector de movimiento predicho entre vistas para el modo de fusión/salto.

15 La FIG. 12 es una tabla que indica una especificación de ejemplo de índices de referencia en 3D-HEVC.

La FIG. 13 es un diagrama conceptual que ilustra una derivación de ejemplo de un candidato heredado de vector de movimiento para codificación de profundidad.

20 La FIG. 14 ilustra la estructura de predicción de la predicción residual avanzada (ARP) en la codificación de vídeo multivista.

La FIG. 15 es un diagrama conceptual que ilustra una relación de ejemplo entre un bloque actual, un bloque de referencia y bloques compensados por movimiento.

25 La FIG. 16 es un diagrama conceptual que ilustra la predicción de movimiento entre vistas de unidad de subpredicción.

30 La FIG. 17 es un dibujo conceptual que representa las técnicas de predicción de síntesis de vista inversa y compensación de movimiento de la presente divulgación cuando se usa particionamiento de movimiento asimétrica.

35 La FIG. 18 es un diagrama conceptual que ilustra las técnicas de herencia de vector de movimiento y compensación de movimiento para tamaños de particionamiento de movimiento asimétrica de 4x16 y 16x4.

La FIG. 19 es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo que puede implementar las técnicas de predicción inter de la presente divulgación.

40 La FIG. 20 es un diagrama de bloques que ilustra un ejemplo de un descodificador de vídeo que puede implementar las técnicas de predicción inter de la presente divulgación.

La FIG. 21 es un diagrama de flujo que ilustra un procedimiento de codificación de ejemplo de la divulgación.

45 La FIG. 22 es un diagrama de flujo que ilustra otro procedimiento de codificación de ejemplo de la divulgación.

La FIG. 23 es un diagrama de flujo que ilustra otro procedimiento de codificación de ejemplo de la divulgación.

50 La FIG. 24 es un diagrama de flujo que ilustra un procedimiento de descodificación de ejemplo de la divulgación.

La FIG. 25 es un diagrama de flujo que ilustra un procedimiento de descodificación de ejemplo de la divulgación.

55 La FIG. 26 es un diagrama de flujo que ilustra un procedimiento de descodificación de ejemplo de la divulgación.

DESCRIPCIÓN DETALLADA

60 **[0014]** En general, la presente divulgación describe técnicas relacionadas con la codificación de vídeo 3D en base a códecs avanzados, incluyendo la codificación de una o más vistas junto con un bloque de profundidad usando el códec 3D-HEVC (codificación de vídeo de alta eficacia). En particular, la presente divulgación describe técnicas para dividir adicionalmente las unidades de predicción (PU) separadas usando técnicas de particionamiento de movimiento asimétrica en subbloques más pequeños. Las técnicas de la presente divulgación incluyen técnicas para derivar y/o heredar vectores de movimiento y vectores de movimiento de disparidad para subbloques de PU separadas usando
65 particionamiento de movimiento asimétrica.

[0015] La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y decodificación de vídeo 10 de ejemplo que puede utilizar técnicas de la presente divulgación. Como se muestra en la FIG. 1, el sistema 10 incluye un dispositivo de origen 12 que proporciona datos de vídeo codificados que se van a decodificar en un momento posterior por un dispositivo de destino 14. En particular, el dispositivo de origen 12 puede proporcionar datos de vídeo al dispositivo de destino 14 por medio de un medio legible por ordenador 16. El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender cualquiera de una amplia gama de dispositivos, incluyendo ordenadores de sobremesa, ordenadores plegables (es decir, portátiles), ordenadores de tableta, decodificadores, equipos telefónicos de mano tales como los denominados teléfonos "inteligentes", los denominados paneles "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, 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 dispositivo de destino 14 puede recibir los datos de vídeo codificados que se van a decodificar por medio del medio legible por ordenador 16. El medio legible por ordenador 16 puede comprender cualquier tipo de medio o dispositivo que puede transportar los datos de vídeo codificados desde el dispositivo de origen 12 al dispositivo de destino 14. En un ejemplo, el medio legible por ordenador 16 puede comprender un medio de comunicación para habilitar el dispositivo de origen 12 para transmitir datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real. Los datos de vídeo codificados se pueden modular de acuerdo con una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitir al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrica o por cable, tal como un espectro de radiofrecuencia (RF) o una o más líneas físicas de transmisión. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área amplia o una red global, tal como Internet. El medio de comunicación puede incluir enrutadores, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo de origen 12 al dispositivo de destino 14.

[0017] En algunos ejemplos, se pueden emitir datos codificados desde la interfaz de salida 22 a un dispositivo de almacenamiento. De forma similar, se puede acceder a los datos codificados desde el dispositivo de almacenamiento por una interfaz de entrada. El dispositivo de almacenamiento puede incluir cualquiera de una variedad de medios de almacenamiento de datos, distribuidos o de acceso local, tales como un disco duro, discos Blu-ray, DVD, CD-ROM, memoria flash, memoria volátil o no volátil o cualquier otro medio de almacenamiento digital adecuado para almacenar datos de vídeo codificados. En otro ejemplo, el dispositivo de almacenamiento puede corresponder a un servidor de ficheros o a otro dispositivo de almacenamiento intermedio que pueda almacenar el vídeo codificado generado por el dispositivo de origen 12. El dispositivo de destino 14 puede acceder a datos de vídeo almacenados desde el dispositivo de almacenamiento por medio de transmisión continua o descarga. El servidor de ficheros puede ser cualquier tipo de servidor que pueda almacenar datos de vídeo codificados y transmitir esos datos de vídeo codificados al dispositivo de destino 14. Los servidores de ficheros de ejemplo incluyen un servidor web (por ejemplo, para un sitio web), un servidor FTP, dispositivos de almacenamiento conectados en red (NAS) o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de cualquier conexión de datos estándar, incluyendo una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión wifi), una conexión por cable (por ejemplo, DSL, módem de cable, etc.) o una combinación de ambas que sea adecuada para acceder a datos de vídeo codificados almacenados en un servidor de ficheros. La transmisión de datos de vídeo codificados desde el dispositivo de almacenamiento puede ser una transmisión continua, una transmisión de descarga o una combinación de ambas.

[0018] Las técnicas de la presente divulgación no están limitadas necesariamente a aplicaciones o contextos inalámbricos. Las técnicas se pueden aplicar a la codificación de vídeo como apoyo de cualquiera de una variedad de aplicaciones de multimedia, tales como radiodifusiones de televisión por el aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones continuas de vídeo por internet, tales como la transmisión continua adaptativa dinámica por HTTP (DASH), vídeo digital que se codifica en un medio de almacenamiento de datos, 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 admitir una transmisión de vídeo unidireccional o bidireccional para admitir aplicaciones tales como transmisión continua de vídeo, reproducción de vídeo, radiodifusión de vídeo y/o videotelefonía.

[0019] En el ejemplo de la FIG. 1, el dispositivo de origen 12 incluye un origen de vídeo 18, una unidad de estimación de profundidad 19, un codificador de vídeo 20 y una interfaz de salida 22. El dispositivo de destino 14 incluye la interfaz de entrada 28, el decodificador de vídeo 30, la unidad de representación basada en la imagen de profundidad (DIBR) 31 y el dispositivo de visualización 32. En otros ejemplos, un dispositivo de origen y un dispositivo de destino pueden incluir otros componentes o disposiciones. Por ejemplo, el dispositivo de origen 12 puede recibir datos de vídeo desde un origen de vídeo externo 18, tal como una cámara externa. Del mismo modo, el dispositivo de destino 14 puede interactuar con un dispositivo de visualización externo, en lugar de incluir un dispositivo de visualización integrado.

[0020] El sistema 10 ilustrado de la FIG. 1 es simplemente un ejemplo. Las técnicas de la presente divulgación se pueden realizar por cualquier dispositivo de codificación y/o decodificación de vídeo digital. Aunque, en general, las técnicas de la presente divulgación se realizan por un dispositivo de codificación de vídeo, las técnicas también se pueden realizar por un codificador/decodificador de vídeo, denominado típicamente "CÓDEC". Además, las técnicas

de la presente divulgación también se pueden realizar por un preprocesador de vídeo. El dispositivo de origen 12 y el dispositivo de destino 14 son simplemente ejemplos de dichos dispositivos de codificación en los que el dispositivo de origen 12 genera datos de vídeo codificados para su transmisión al dispositivo de destino 14. En algunos ejemplos, los dispositivos 12, 14 pueden funcionar de manera sustancialmente simétrica, de modo que cada uno de los dispositivos 12, 14 incluya componentes de codificación y decodificación de vídeo. De ahí que el sistema 10 pueda admitir la transmisión de vídeo unidireccional o bidireccional entre los dispositivos de vídeo 12, 14, por ejemplo, para la transmisión continua de vídeo, la reproducción de vídeo, la radiodifusión de vídeo o la videotelefonía.

[0021] El origen de vídeo 18 del dispositivo de origen 12 puede incluir un dispositivo de captación de vídeo, tal como una videocámara, un archivo de vídeo que contiene vídeo captado previamente y/o una interfaz de alimentación de vídeo para recibir vídeo desde un proveedor de contenidos de vídeo. Como otra alternativa, el origen de vídeo 18 puede generar datos basados en gráficos por ordenador como la origen de vídeo, o una combinación de vídeo en directo, vídeo archivado y vídeo generado por ordenador. En algunos casos, si el origen de vídeo 18 es una cámara de vídeo, el dispositivo de origen 12 y el dispositivo de destino 14 pueden formar los denominados teléfonos con cámara o videoteléfonos. Sin embargo, como se menciona anteriormente, las técnicas descritas en la presente divulgación pueden ser aplicables a la codificación de vídeo en general, y se pueden aplicar a aplicaciones inalámbricas y/o por cable. En cada caso, el vídeo captado, precaptado o generado por ordenador se puede codificar por el codificador de vídeo 20. A continuación, la información de vídeo codificada se puede emitir por la interfaz de salida 22 a un medio legible por ordenador 16.

[0022] El origen de vídeo 18 puede proporcionar una o más vistas de datos de vídeo al codificador de vídeo 20. Por ejemplo, el origen de vídeo 18 puede corresponder a una formación de cámaras, teniendo cada una una posición horizontal única con respecto a una escena particular que se está filmando. De forma alternativa, el origen de vídeo 18 puede generar datos de vídeo desde perspectivas dispares de cámara horizontal, por ejemplo, usando gráficos por ordenador. La unidad de estimación de profundidad 19 se puede configurar para determinar valores para píxeles de profundidad correspondientes a píxeles en una imagen de textura. Por ejemplo, la unidad de estimación de profundidad 19 puede representar una unidad de navegación y medición por sonido (SONAR), una unidad de detección y medición por luz (LIDAR) u otra unidad que pueda determinar directamente los valores de profundidad de forma sustancialmente simultánea mientras que graba datos de vídeo de una escena.

[0023] Adicionalmente o de forma alternativa, la unidad de estimación de profundidad 19 se puede configurar para calcular valores de profundidad de manera indirecta comparando dos o más imágenes que se captaron sustancialmente al mismo tiempo desde diferentes perspectivas de cámara horizontal. Calculando la disparidad horizontal entre valores de píxeles sustancialmente similares en las imágenes, la unidad de estimación de profundidad 19 se puede aproximar a la profundidad de diversos objetos en la escena. La unidad de estimación de profundidad 19 puede estar integrada funcionalmente con el origen de vídeo 18, en algunos ejemplos. Por ejemplo, cuando el origen de vídeo 18 genera imágenes de gráficos por ordenador, la unidad de estimación de profundidad 19 puede proporcionar mapas de profundidad reales para objetos gráficos, por ejemplo, usando coordenadas z de píxeles y objetos usados para representar imágenes de textura.

[0024] El medio legible por ordenador 16 puede incluir medios transitorios, tales como una radiodifusión inalámbrica o transmisión de red por cable, o medios de almacenamiento (es decir, medios de almacenamiento no transitorios), tales como un disco duro, una unidad de memoria flash, un disco compacto, un disco de vídeo digital, un disco Blu-ray u otros medios legibles por ordenador. En algunos ejemplos, un servidor de red (no mostrado) puede recibir datos de vídeo codificados desde el dispositivo de origen 12 y proporcionar los datos de vídeo codificados al dispositivo de destino 14, por ejemplo, por medio de transmisión por red. De forma similar, un dispositivo informático de una instalación de producción de un medio, tal como una instalación de acuñación de discos, puede recibir datos de vídeo codificados desde el dispositivo de origen 12 y producir un disco que contiene los datos de vídeo codificados. Por lo tanto, se puede entender que el medio legible por ordenador 16 incluye uno o más medios legibles por ordenador de diversas formas, en diversos ejemplos.

[0025] La interfaz de entrada 28 del dispositivo de destino 14 recibe información desde el medio legible por ordenador 16. La información del medio legible por ordenador 16 puede incluir información sintáctica definida por el codificador de vídeo 20, que también se usa por el decodificador de vídeo 30, que incluye elementos sintácticos que describen características y/o procesamiento de bloques y otras unidades codificadas, por ejemplo, los GOP. El dispositivo de visualización 32 visualiza los datos de vídeo descodificados a un usuario, y puede comprender cualquiera de una variedad de dispositivos de visualización, tales como un tubo de rayos catódicos (CRT), una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de visualización. En algunos ejemplos, el dispositivo de visualización 32 puede comprender un dispositivo que puede visualizar dos o más vistas de forma simultánea o de forma sustancialmente simultánea, por ejemplo, para producir un efecto visual 3D para un observador.

[0026] La unidad de DIBR 31 del dispositivo de destino 14 puede representar vistas sintetizadas usando información de textura y profundidad de vistas descodificadas, recibidas desde el decodificador de vídeo 30. Por ejemplo, la unidad de DIBR 31 puede determinar la disparidad horizontal para datos de píxeles de imágenes de textura como una función de los valores de píxeles en mapas de profundidad correspondientes. A continuación, la unidad de DIBR 31

puede generar una imagen sintetizada desplazando píxeles en una imagen de textura, a la izquierda o a la derecha, en la disparidad horizontal determinada. De esta manera, el dispositivo de visualización 32 puede visualizar una o más vistas, que pueden corresponder a vistas descodificadas y/o vistas sintetizadas, en cualquier combinación. De acuerdo con las técnicas de la presente divulgación, el descodificador de vídeo 30 puede proporcionar valores de precisión originales y actualizados para los rangos de profundidad y parámetros de cámara a la unidad DIBR 31, que puede usar los rangos de profundidad y los parámetros de cámara para sintetizar apropiadamente las vistas.

[0027] Aunque no se muestra en la FIG. 1, en algunos aspectos, el codificador de vídeo 20 y el descodificador de vídeo 30 se pueden integrar, cada uno, en un codificador y descodificador de audio, y pueden incluir unidades MUX-DEMUX apropiadas, u otro hardware y software, para tratar la codificación tanto de audio como de vídeo en un flujo de datos común o en flujos de datos separados. Si fuera aplicable, las unidades MUX-DEMUX se pueden ajustar al protocolo multiplexor ITU H.223, o a otros protocolos tales como el protocolo de datagramas de usuario (UDP).

[0028] El codificador de vídeo 20 y el descodificador de vídeo 30 se pueden implementar cada uno como cualquiera de una variedad de circuitos codificadores adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), matrices de puertas programables *in situ* (FPGA), lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. Cuando las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio no transitorio legible por ordenador adecuado y ejecutar las instrucciones en hardware usando uno o más procesadores para realizar las técnicas de la presente divulgación. Tanto el codificador de vídeo 20 como el descodificador de vídeo 30 se pueden incluir en uno o más codificadores o descodificadores, de los que cualquiera se puede integrar como parte de un codificador/descodificador (CÓDEC) combinado en un dispositivo respectivo. Un dispositivo que incluye el codificador de vídeo 20 y/o el descodificador de vídeo 30 puede comprender un circuito integrado, un microprocesador y/o un dispositivo de comunicación inalámbrica, tal como un teléfono móvil.

[0029] El codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con una norma de codificación de vídeo, tal como la norma de codificación de vídeo de alta eficacia (HEVC), actualmente en fase de desarrollo, y se pueden ajustar al modelo de prueba de la HEVC (HM). De forma alternativa, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con otras normas privadas o industriales, tales como la norma ITU-T H.264, de forma alternativa denominada MPEG-4, parte 10, codificación de vídeo avanzada (AVC), o extensiones de dichas normas, tales como la extensión MVC de ITU-T H.264/AVC. El más reciente borrador conjunto de la MVC se describe en "Advanced video coding for generic audiovisual", Recomendación H.264 de la ITU-T, marzo de 2010. En particular, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con la norma de codificación 3D y/o multivista, incluyendo una extensión 3D de la norma HEVC (por ejemplo, 3D-HEVC).

[0030] Un borrador de la norma HEVC, denominado "HEVC Working Draft 10" o "WD10", se describe en el documento JCTVC-L1003v34, de Bross *et al.*, "High efficiency video coding (HEVC) text specification draft 10 (for FDIS & Last Call)", Equipo de Colaboración Conjunta sobre Codificación de Vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 12.^a Conferencia: Ginebra, Suiza, 14-23 de enero de 2013, que, a partir del 22 de agosto de 2014, se puede descargar desde: http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v34.zip.

[0031] Otro borrador de la norma HEVC, se denomina en el presente documento "WD10 revisions", descrito en Bross *et al.*, "Editors' proposed corrections to HEVC version 1", Equipo de Colaboración Conjunta sobre Codificación de Vídeo (JCT-VC) de UIT-T SG16 WP3 e ISO/IEC JTC 1/SC29/WG11, 13.^a Conferencia, Incheon, KR, abril de 2013, que, a partir del 22 de agosto de 2014, está disponible en: http://phenix.int-evry.fr/jct/doc_end_user/documents/13_Incheon/wg11/JCTVC-M0432-v3.zip. Una extensión multivista de HEVC, a saber, la MV-HEVC también se está desarrollando por el JCT-3V.

[0032] Actualmente, un Equipo de Colaboración Conjunta sobre Codificación de Vídeo 3D (JCT-3C) de VCEG y MPEG está desarrollando una norma 3DV en base a la HEVC, para la que parte de los trabajos de normalización incluye la normalización del códec de vídeo multivista en base a la HEVC (MV-HEVC) y otra parte para la codificación de vídeo 3D en base a la HEVC (3D-HEVC). Para MV-HEVC, se debe garantizar que solo haya cambios sintácticos de alto nivel (HLS) en la misma, de modo que no sea necesario rediseñar ningún módulo a nivel de la unidad de codificación/unidad de predicción en HEVC y se pueda reutilizar por completo para MV-HEVC. Para 3D-HEVC, se pueden incluir y admitir nuevas herramientas de codificación, incluyendo aquellas a nivel de unidad de codificación/unidad de predicción, tanto para las vistas de textura como de profundidad.

[0033] Un software de versión 3D-HTM para 3D-HEVC se puede descargar desde el siguiente enlace: [3D-HTM versión 8.0]: https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSsoftware/tags/HTM-8.0/. Un borrador de trabajo de 3D-HEVC (número de documento: E1001) está disponible en: http://phenix.it-sudparis.eu/jct2/doc_end_user/current_document.php?id=1361. La más reciente descripción del software (número de documento: E1005) está disponible en: http://phenix.it-sudparis.eu/ict2/doc_end_user/current_document.php?id=1360.

[0034] Una versión más reciente del software de 3D-HTM para 3D-HEVC se puede descargar desde el siguiente enlace: [3D-HTM versión 12.0]: https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSsoftware/tags/HTM-12.0/. El borrador de trabajo correspondiente de 3D-HEVC (número de documento: 11001) está disponible en: http://phenix.int-evry.fr/ict3v/doc_end_user/current_document.php?id=2299. La más reciente descripción del software (número de documento: 11005) está disponible en: http://phenix.int-evry.fr/jct3v/doc_end_user/current_document.php?id=2301.

[0035] Inicialmente, se analizarán técnicas de codificación de ejemplo de HEVC. Los trabajos de normalización de HEVC se basaron en un modelo en evolución de un dispositivo de codificación de vídeo denominado modelo de prueba de HEVC (HM). El HM suponía varias capacidades adicionales de los dispositivos de codificación de vídeo con respecto a los dispositivos existentes, de acuerdo con, por ejemplo, ITU-T H.264/AVC. Por ejemplo, mientras que H.264 proporciona nueve modos de codificación por predicción intra, el HM puede proporcionar hasta treinta y tres modos de codificación por predicción intra angulares más modos DC y planos.

[0036] En HEVC y otras especificaciones de codificación de vídeo, una secuencia de vídeo incluye típicamente una serie de imágenes. Las imágenes también se pueden denominar "tramas". Una imagen puede incluir tres matrices de muestras, indicadas como S_L , S_{Cb} y S_{Cr} . S_L es una matriz bidimensional (es decir, un bloque) de muestras de luma. S_{Cb} es una matriz bidimensional de muestras de crominancia Cb. S_{Cr} es una matriz bidimensional de muestras de crominancia Cr. Las muestras de crominancia también se pueden denominar en el presente documento muestras de "croma". En otros casos, una imagen puede ser monocromática y puede incluir solo una matriz de muestras de luma.

[0037] Para generar una representación codificada de una imagen, el codificador de vídeo 20 puede generar un conjunto de unidades de árbol de codificación (CTU). Cada una de las CTU puede comprender un bloque de árbol de codificación de muestras de luma, dos bloques de árbol de codificación correspondientes de muestras de croma y unas estructuras sintácticas usadas para codificar las muestras de los bloques de árbol de codificación. En imágenes monocromáticas o imágenes que tienen tres planos de color separados, una CTU puede comprender un único bloque de árbol de codificación y unas estructuras sintácticas usadas para codificar las muestras del bloque de árbol de codificación. Un bloque de árbol de codificación puede ser un bloque de muestras $N \times N$. Una CTU también se puede denominar "bloque de árbol" o "unidad de codificación de máximo tamaño" (LCU). Las CTU de la HEVC pueden ser ampliamente análogas a los macrobloques de otras normas, tales como H.264/AVC. Sin embargo, una CTU no está necesariamente limitada a un tamaño particular y puede incluir una o más unidades de codificación (CU). Un fragmento puede incluir un número entero de CTU ordenadas consecutivamente en un orden de exploración de barrido.

[0038] Para generar una CTU codificada, el codificador de vídeo 20 puede realizar de forma recursiva una particionamiento de árbol cuádruple en los bloques de árbol de codificación de una CTU para dividir los bloques de árbol de codificación en bloques de codificación, de ahí el nombre "unidades de árbol de codificación". Un bloque de codificación es un bloque de muestras $N \times N$. Una unidad de codificación (CU) puede comprender un bloque de codificación de muestras de luma y dos bloques de codificación correspondientes de muestras de croma de una imagen que tenga una matriz de muestras de luma, una matriz de muestras de Cb y una matriz de muestras de Cr, y estructuras sintácticas usadas para codificar las muestras de los bloques de codificación. En imágenes monocromáticas o imágenes que tienen tres planos de color separados, una CU puede comprender un único bloque de codificación y estructuras sintácticas usadas para codificar las muestras del bloque de codificación.

[0039] El codificador de vídeo 20 puede particionar un bloque de codificación de una CU en uno o más bloques de predicción. Un bloque de predicción es un bloque rectangular (es decir, cuadrado o no cuadrado) de muestras en las que se aplica la misma predicción. Una unidad de predicción (PU) de una CU puede comprender un bloque de predicción de muestras de luma, dos bloques de predicción correspondientes de muestras de croma y estructuras sintácticas usadas para predecir los bloques de predicción. En imágenes monocromáticas o imágenes que tienen tres planos de color separados, una PU puede comprender un único bloque de predicción y estructuras sintácticas usadas para predecir el bloque de predicción. El codificador de vídeo 20 puede generar bloques predictivos de luma, Cb y Cr para bloques de predicción de luma, Cb y Cr de cada PU de la CU.

[0040] El codificador de vídeo 20 puede usar predicción intra o predicción inter para generar los bloques predictivos para una PU. Si el codificador de vídeo 20 usa predicción intra para generar los bloques predictivos de una PU, el codificador de vídeo 20 puede generar los bloques predictivos de la PU en base a muestras descodificadas de la imagen asociada con la PU. En algunas versiones de HEVC, para el componente de luma de cada PU, se utiliza un procedimiento de predicción intra con 33 modos de predicción angular (indexados de 2 a 34), modo DC (indexado con 1) y modo plano (indexado con 0).

[0041] Si el codificador de vídeo 20 utiliza predicción inter para generar los bloques predictivos de una PU, el codificador de vídeo 20 puede generar los bloques predictivos de la PU en base a muestras descodificadas de una o más imágenes distintas a la imagen asociada con la PU. La predicción inter puede ser predicción inter unidireccional (es decir, unipredicción o predicción unipredictiva) o predicción inter bidireccional (es decir, bipredicción o predicción bipredictiva). Para realizar unipredicción o la bipredicción, el codificador de vídeo 20 puede generar una primera lista de imágenes de referencia (RefPicList0) y una segunda lista de imágenes de referencia (RefPicList1) para un fragmento actual. Cada una de las listas de imágenes de referencia puede incluir una o más imágenes de referencia. Cuando se usa unipredicción, el codificador de vídeo 20 puede buscar las imágenes de referencia en cualquiera de o

en ambas RefPicList0 y RefPicList1 para determinar una ubicación de referencia dentro de una imagen de referencia. Además, cuando se usa unipredicción, el codificador de vídeo 20 puede generar, en base, al menos en parte, a las muestras correspondientes a la ubicación de referencia, los bloques de muestras predictivos para la PU. Además, cuando se usa unipredicción, el codificador de vídeo 20 puede generar un único vector de movimiento que indica un desplazamiento espacial entre un bloque de predicción de la PU y la ubicación de referencia. Para indicar el desplazamiento espacial entre un bloque de predicción de la PU y la ubicación de referencia, un vector de movimiento puede incluir una componente horizontal que especifica un desplazamiento horizontal entre el bloque de predicción de la PU y la ubicación de referencia y puede incluir una componente vertical que especifica un desplazamiento vertical entre el bloque de predicción de la PU y la ubicación de referencia.

[0042] Cuando se usa la bipredicción para codificar una PU, el codificador de vídeo 20 puede determinar una primera ubicación de referencia en una imagen de referencia en RefPicList0 y una segunda ubicación de referencia en una imagen de referencia en RefPicList1. A continuación, el codificador de vídeo 20 puede generar, en base, al menos en parte, a muestras correspondientes a las primera y segunda ubicaciones de referencia, los bloques predictivos para la PU. Además, cuando se usa bipredicción para codificar la PU, el codificador de vídeo 20 puede generar un primer vector de movimiento que indica un desplazamiento espacial entre un bloque de muestra de la PU y la primera ubicación de referencia y un segundo vector de movimiento que indica un desplazamiento espacial entre el bloque de predicción de la PU y la segunda ubicación de referencia.

[0043] Típicamente, una construcción de lista de imágenes de referencia para la primera o la segunda lista de imágenes de referencia (por ejemplo, RefPicList0 o RefPicList1) de una imagen B incluye dos etapas: inicialización de la lista de imágenes de referencia y reordenamiento de la lista de imágenes de referencia (modificación). La inicialización de la lista de imágenes de referencia es un mecanismo explícito que coloca las imágenes de referencia en la memoria de imágenes de referencia (también conocida como memoria intermedia de imágenes descodificadas) en una lista en base al orden de los valores POC (recuento de orden de imágenes, alineado con el orden de visualización de una imagen). El mecanismo de reordenamiento de la lista de imágenes de referencia puede modificar la posición de una imagen que se colocó en la lista durante la inicialización de la lista de imágenes de referencia en cualquier posición nueva, o colocar cualquier imagen de referencia de la memoria de imágenes de referencia en cualquier posición incluso si la imagen no pertenece a la lista inicializada. Algunas imágenes después del reordenamiento (modificación) de la lista de imágenes de referencia, se pueden colocar en una posición mucho más lejana en la lista. Sin embargo, si una posición de una imagen supera el número de imágenes de referencia activas de la lista, la imagen no se considera como una entrada de la lista de imágenes de referencia final. El número de imágenes de referencia activas se puede señalar en la cabecera de fragmento para cada lista.

[0044] Después de que se construyen las listas de imágenes de referencia (a saber RefPicList0 y RefPicList1, si están disponibles), se puede usar un índice de referencia a una lista de imágenes de referencia para identificar cualquier imagen de referencia incluida en la lista de imágenes de referencia.

[0045] Después de que el codificador de vídeo 20 genera bloques de luma, Cb y Cr predictivos para una o más PU de una CU, el codificador de vídeo 20 puede generar un bloque residual de luma para la CU. Cada muestra del bloque residual de luma de la CU indica una diferencia entre una muestra de luma de uno de los bloques de luma predictivos de la CU y una muestra correspondiente del bloque de codificación de luma original de la CU. Además, el codificador de vídeo 20 puede generar un bloque residual de Cb para la CU. Cada muestra del bloque residual de Cb de la CU puede indicar una diferencia entre una muestra de Cb de uno de los bloques de Cb predictivos de la CU y una muestra correspondiente del bloque de codificación de Cb original de la CU. El codificador de vídeo 20 también puede generar un bloque residual de Cr para la CU. Cada muestra del bloque residual de Cr de la CU puede indicar una diferencia entre una muestra de Cr de uno de los bloques de Cr predictivos de la CU y una muestra correspondiente del bloque de codificación de Cr original de la CU.

[0046] Además, el codificador de vídeo 20 puede usar particionamiento de árbol cuádruple para descomponer los bloques residuales de luma, Cb y Cr de una CU en uno o más bloques de transformada de luma, Cb y Cr. Un bloque de transformada es un bloque rectangular (por ejemplo, cuadrado o no cuadrado) de muestras a las que se aplica la misma transformada. Una unidad de transformada (TU) de una CU puede comprender un bloque de transformada de muestras de luma, dos bloques de transformada correspondientes de muestras de croma y estructuras sintácticas usadas para transformar las muestras de bloques de transformada. Por tanto, cada TU de una CU puede estar asociada con un bloque de transformada de luma, un bloque de transformada de Cb y un bloque de transformada de Cr. El bloque de transformada de luma asociado con la TU puede ser un subbloque del bloque residual de luma de la CU. El bloque de transformada de Cb puede ser un subbloque del bloque residual de Cb de la CU. El bloque de transformada de Cr puede ser un subbloque del bloque residual de Cr de la CU. En imágenes monocromáticas o imágenes que tienen tres planos de color separados, una TU puede comprender un único bloque de transformada y estructuras sintácticas usadas para transformar las muestras del bloque de transformada.

[0047] El codificador de vídeo 20 puede aplicar una o más transformadas a un bloque de transformada de luma de una TU para generar un bloque de coeficientes de luma para la TU. Un bloque de coeficientes puede ser una matriz bidimensional de coeficientes de transformada. Un coeficiente de transformada puede ser una cantidad escalar. El codificador de vídeo 20 puede aplicar una o más transformadas a un bloque de transformada de Cb de una TU para

generar un bloque de coeficientes de Cb para la TU. El codificador de vídeo 20 puede aplicar una o más transformadas a un bloque de transformada de Cr de una TU para generar un bloque de coeficientes de Cr para la TU.

5 **[0048]** Después de generar un bloque de coeficientes (por ejemplo, un bloque de coeficientes de luma, un bloque de coeficientes de Cb o un bloque de coeficientes de Cr), el codificador de vídeo 20 puede cuantificar el bloque de coeficientes. La cuantificación se refiere, en general, a un procedimiento en el que coeficientes de transformada se cuantifican para reducir, posiblemente, la cantidad de datos usados para representar los coeficientes de transformada, proporcionando más compresión. Después de que el codificador de vídeo 20 cuantifica un bloque de coeficientes, el codificador de vídeo 20 puede codificar por entropía elementos sintácticos que indican los coeficientes de transformada
10 cuantificados. Por ejemplo, el codificador de vídeo 20 puede realizar una codificación aritmética binaria adaptativa al contexto (CABAC) en los elementos sintácticos que indican los coeficientes de transformada cuantificados.

15 **[0049]** El codificador de vídeo 20 puede emitir un flujo de bits que incluye una secuencia de bits que forma una representación de imágenes codificadas y datos asociados. El flujo de bits puede comprender una secuencia de unidades de capa de abstracción de red (NAL). Una unidad de NAL es una estructura sintáctica que contiene una indicación del tipo de datos en la unidad de NAL y los octetos que contienen esos datos en forma de una carga útil de secuencia de octetos sin procesar (RBSP) entremezclados como sea necesario con bits de prevención de emulación. Cada una de las unidades de NAL incluye una cabecera de unidad de NAL y encapsula una RBSP. La cabecera de unidad de NAL puede incluir un elemento sintáctico que indica un código de tipo de unidad de NAL. El código de tipo
20 de unidad de NAL especificado por la cabecera de unidad de NAL de una unidad de NAL indica el tipo de la unidad de NAL. Una RBSP puede ser una estructura sintáctica que contiene un número entero de octetos que están encapsulados dentro de una unidad de NAL. En algunos casos, una RBSP incluye bits cero.

25 **[0050]** Diferentes tipos de unidades de NAL pueden encapsular diferentes tipos de RBSP. Por ejemplo, un primer tipo de unidad de NAL puede encapsular una RBSP para un conjunto de parámetros de imagen (PPS), un segundo tipo de unidad de NAL puede encapsular una RBSP para un fragmento codificado, un tercer tipo de unidad de NAL puede encapsular una RBSP para SEI y así sucesivamente. Las unidades de NAL que encapsulan las RBSP para datos de codificación de vídeo (a diferencia de las RBSP para conjuntos de parámetros y mensajes SEI) se pueden denominar unidades de NAL de capa de codificación de vídeo (VCL).
30

35 **[0051]** El descodificador de vídeo 30 puede recibir un flujo de bits generado por el codificador de vídeo 20. Además, el descodificador de vídeo 30 puede analizar sintácticamente el flujo de bits para obtener elementos sintácticos a partir del flujo de bits. El descodificador de vídeo 30 puede reconstruir las imágenes de los datos de vídeo en base, al menos en parte, a los elementos sintácticos obtenidos a partir del flujo de bits. El procedimiento para reconstruir los datos de vídeo puede ser, en general, recíproco al procedimiento realizado por el codificador de vídeo 20. Por ejemplo, el descodificador de vídeo 30 puede usar vectores de movimiento de las PU para determinar bloques predictivos para las PU de una CU actual. Además, el descodificador de vídeo 30 puede cuantificar inversamente bloques de coeficientes asociados con las TU de la CU actual. El descodificador de vídeo 30 puede realizar transformadas inversas en los bloques de coeficientes para reconstruir bloques de transformada asociados con las TU de la CU actual. El descodificador de vídeo 30 puede reconstruir los bloques de codificación de la CU actual añadiendo las muestras de los bloques predictivos para las PU de la CU actual a muestras correspondientes de los bloques de transformada de las TU de la CU actual. Reconstruyendo los bloques de codificación para cada CU de una imagen, el descodificador de vídeo 30 puede reconstruir la imagen.
40

45 **[0052]** En algunos ejemplos, el codificador de vídeo 20 puede señalar la información de movimiento de una PU usando el modo de fusión (*merge*) o el modo de predicción de vector de movimiento avanzado (AMVP). En otras palabras, en HEVC, existen dos modos para la predicción de los parámetros de movimiento, uno es el modo de fusión y el otro es AMVP. La predicción de movimiento puede comprender la determinación de la información de movimiento de una unidad de vídeo (por ejemplo, una PU) en base a la información de movimiento de una o más de otras unidades de vídeo. La información de movimiento de una PU puede incluir el/los vector(es) de movimiento de la PU y el/los índice(s) de referencia de la PU.
50

55 **[0053]** Cuando el codificador de vídeo 20 señala la información de movimiento de una PU actual usando el modo de fusión, el codificador de vídeo 20 genera una lista de candidatos de fusión. En otras palabras, el codificador de vídeo 20 puede realizar un procedimiento de construcción de lista de predictores de vector de movimiento. La lista de candidatos de fusión incluye un conjunto de candidatos de fusión que indican la información de movimiento de las PU que están contiguas espacial o temporalmente a la PU actual. Es decir, en el modo de fusión, se construye una lista de candidatos de parámetros de movimiento (por ejemplo, índices de referencia, vectores de movimiento, etc.) donde un candidato puede ser de bloques contiguos espaciales y temporales. En algunos ejemplos, los candidatos también pueden incluir un candidato generado artificialmente.
60

65 **[0054]** Además, en el modo de fusión, el codificador de vídeo 20 puede seleccionar un candidato de fusión de la lista de candidatos de fusión y puede usar la información de movimiento indicada por el candidato de fusión seleccionada como la información de movimiento de la PU actual. El codificador de vídeo 20 puede señalar la posición en la lista de candidatos de fusión del candidato de fusión seleccionado. Por ejemplo, el codificador de vídeo 20 puede señalar los parámetros de vector de movimiento seleccionados transmitiendo un índice a la lista de candidatos. El descodificador

de vídeo 30 puede obtener, del flujo de bits, el índice en la lista de candidatos (es decir, un índice de lista de candidatos). Además, el descodificador de vídeo 30 puede generar la misma lista de candidatos de fusión y puede determinar, en base a la indicación de la posición del candidato de fusión seleccionado, el candidato de fusión seleccionado. A continuación, el descodificador de vídeo 30 puede usar la información de movimiento del candidato de fusión seleccionado para generar bloques predictivos para la PU actual. Es decir, el descodificador de vídeo 30 puede determinar, en base, al menos en parte, al índice de lista de candidatos, un candidato seleccionado en la lista de candidatos, en la que el candidato seleccionado especifica el vector de movimiento para la PU actual. De esta manera, en el lado del descodificador, una vez que se descodifica el índice, todos los parámetros de movimiento del bloque correspondiente donde el índice apunta se pueden heredar por la PU actual.

[0055] El modo de salto es similar al modo de fusión. En el modo de salto, el codificador de vídeo 20 y el descodificador de vídeo 30 generan y usan una lista de candidatos de fusión de la misma manera que el codificador de vídeo 20 y el descodificador de vídeo 30 usan la lista de candidatos de fusión en el modo de fusión. Sin embargo, cuando el codificador de vídeo 20 señala la información de movimiento de una PU actual usando el modo de salto, el codificador de vídeo 20 no señala ningún dato residual para la PU actual. En consecuencia, el descodificador de vídeo 30 puede determinar, sin el uso de datos residuales, un bloque predictivo para la PU en base a un bloque de referencia indicado por la información de movimiento de un candidato seleccionado en la lista de candidatos de fusión.

[0056] El modo AMVP es similar al modo de fusión en que el codificador de vídeo 20 puede generar una lista de candidatos y puede seleccionar un candidato de la lista de candidatos. Sin embargo, cuando el codificador de vídeo 20 señala la información de movimiento RefPicListX de una PU actual usando el modo AMVP, el codificador de vídeo 20 puede señalar una diferencia de vector de movimiento (MVD) RefPicListX para la PU actual y un índice de referencia RefPicListX para la PU actual además de señalar un indicador de MVP RefPicListX para la PU actual. El indicador de MVP RefPicListX para la PU actual puede indicar la posición de un candidato de AMVP seleccionado en la lista de candidatos de AMVP. La MVD RefPicListX para la PU actual puede indicar una diferencia entre un vector de movimiento RefPicListX de la PU actual y un vector de movimiento del candidato de AMVP seleccionado. De esta manera, el codificador de vídeo 20 puede señalar la información de movimiento RefPicListX de la PU actual señalando un indicador de predictor de vector de movimiento (MVP) RefPicListX, un valor de índice de referencia RefPicListX y una MVD RefPicListX. En otras palabras, los datos en el flujo de bits que representan el vector de movimiento para la PU actual pueden incluir datos que representan un índice de referencia, un índice para una lista de candidatos y una MVD.

[0057] Además, cuando la información de movimiento de una PU actual se señala usando el modo AMVP, el descodificador de vídeo 30 puede obtener, del flujo de bits, una MVD para una PU actual y un indicador de MVP. El descodificador de vídeo 30 puede generar la misma lista de candidatos de AMVP y puede determinar, en base al indicador de MVP, el candidato de AMVP seleccionado. El descodificador de vídeo 30 puede recuperar un vector de movimiento de la PU actual añadiendo la MVD al vector de movimiento indicado por el candidato de AMVP seleccionado. Es decir, el descodificador de vídeo 30 puede determinar, en base a un vector de movimiento indicado por el candidato de AMVP seleccionado y la MVD, el vector de movimiento de la PU actual. A continuación, el descodificador de vídeo 30 puede usar el vector de movimiento recuperado o los vectores de movimiento de la PU actual para generar bloques predictivos para la PU actual.

[0058] Cuando el descodificador de vídeo 30 genera una lista de candidatos de AMVP para una PU actual, el descodificador de vídeo 30 puede derivar uno o más candidatos de AMVP en base a la información de movimiento de las PU (es decir, las PU contiguas espacialmente) que abarcan ubicaciones que están contiguas espacialmente a la PU actual. Una PU puede abarcar una ubicación cuando un bloque de predicción de la PU incluye la ubicación.

[0059] Un candidato en una lista de candidatos de fusión o una lista de candidatos de AMVP que se basa en la información de movimiento de una PU que está contigua temporalmente a una PU actual (es decir, una PU que está en una instancia de tiempo diferente al de la PU actual) se puede denominar TMVP. Es decir, se puede usar un TMVP para mejorar la eficacia de codificación de HEVC y, a diferencia de otras herramientas de codificación, el TMVP puede necesitar acceder al vector de movimiento de una trama en una memoria intermedia de imágenes descodificadas, más específicamente en una lista de imágenes de referencia.

[0060] El uso de los TMVP se puede habilitar o deshabilitar de CVS en CVS (secuencia de vídeo codificada), de fragmento en fragmento o de otro modo. Un elemento sintáctico (por ejemplo, `sps_temporal_mvp_enable_flag`) en un SPS puede indicar si el uso de los TMVP está habilitado para una CVS. Además, cuando el uso de los TMVP está habilitado para una CVS, se puede habilitar o deshabilitar el uso de los TMVP para fragmentos particulares dentro de la CVS. Por ejemplo, un elemento sintáctico (por ejemplo, `slice_temporal_mvp_enable_flag`) en una cabecera de fragmento puede indicar si el uso de los TMVP está habilitado para un fragmento. Por tanto, en un fragmento con predicción inter, cuando el TMVP está habilitado para una CVS completa (por ejemplo, `sps_temporal_mvp_enable_flag` en un SPS se establece en 1), se señala `slice_temporal_mvp_enable_flag` en la cabecera del fragmento para indicar si el TMVP está habilitado para el fragmento actual.

[0061] Para determinar un TMVP, un codificador de vídeo puede identificar en primer lugar una imagen de referencia que incluye una PU que está coubicada, con la PU actual. En otras palabras, el codificador de vídeo puede identificar

una imagen coubicada. Si el fragmento actual de la imagen actual es un fragmento B (es decir, un fragmento que puede incluir las PU con predicción inter bidireccional), el codificador de vídeo 20 puede señalar, en una cabecera de fragmento, un elemento sintáctico (por ejemplo, `collocated_from_I0_flag`) que indica si la imagen coubicada proviene de RefPicList0 o RefPicList1. En otras palabras, cuando el uso de los TMVP está habilitado para un fragmento actual, y el fragmento actual es un fragmento B (por ejemplo, un fragmento que puede incluir las PU con predicción inter bidireccional), el codificador de vídeo 20 puede señalar un elemento sintáctico (por ejemplo, `collocated_from_I0_flag`) en una cabecera de fragmento para indicar si la imagen coubicada está en RefPicList0 o RefPicList1. En otras palabras, para obtener un TMVP, en primer lugar se ha de identificar una imagen coubicada. Si la imagen actual es un fragmento B, se señala un `collocated_from_I0_flag` en la cabecera de fragmento para indicar si la imagen coubicada proviene de RefPicList0 o RefPicList1.

[0062] Después de que el descodificador de vídeo 30 identifica la lista de imágenes de referencia que incluye la imagen coubicada, el descodificador de vídeo 30 puede usar otro elemento sintáctico (por ejemplo, `collocated_ref_idx`), que se puede señalar en una cabecera de fragmento, para identificar una imagen (es decir, la imagen coubicada) en la lista de imágenes de referencia identificada. Es decir, después de que se identifica una lista de imágenes de referencia, se usa `collocated_ref_idx` señalada en una cabecera de fragmento para identificar la imagen en la lista de imágenes de referencia.

[0063] El codificador de vídeo puede identificar una PU coubicada verificando la imagen coubicada. El TMVP puede indicar la información de movimiento de una PU inferior derecha de la CU que contiene la PU coubicada, o bien la información de movimiento de la PU inferior derecha dentro de las PU centrales de la CU que contiene esta PU. Por tanto, se usa el movimiento de la PU inferior derecha de la CU que contiene esta PU, o bien el movimiento de la PU inferior derecha dentro de las PU centrales de la CU que contiene esta PU. La PU inferior derecha de la CU que contiene la PU coubicada puede ser una PU que abarca una ubicación inmediatamente debajo y a la derecha de una muestra inferior derecha de un bloque de predicción de la PU. En otras palabras, el TMVP puede indicar la información de movimiento de una PU que está en la imagen de referencia y que abarca una ubicación que está coubicada con una esquina derecha inferior de la PU actual, o el TMVP puede indicar la información de movimiento de una PU que está en la imagen de referencia y que abarca una ubicación que está coubicada con un centro de la PU actual.

[0064] Cuando los vectores de movimiento identificados por el procedimiento anterior (es decir, los vectores de movimiento de un TMVP) se usan para generar un candidato de movimiento para el modo de fusión o el modo AMVP, el codificador de vídeo puede ajustar a escala los vectores de movimiento en base a la ubicación temporal (reflejada por el valor POC). Por ejemplo, un codificador de vídeo puede incrementar la magnitud de un vector de movimiento en cantidades mayores cuando una diferencia entre los valores POC de una imagen actual y una imagen de referencia es mayor que cuando una diferencia entre los valores POC de la imagen actual y la imagen de referencia es menor.

[0065] El índice de referencia objetivo de todas las posibles listas de imágenes de referencia para el candidato de fusión temporal derivado de un TMVP se puede establecer siempre en 0. Sin embargo, para AMVP, el índice de referencia objetivo de todas las posibles imágenes de referencia se establece igual al índice de referencia descodificado. En otras palabras, el índice de referencia objetivo de todas las posibles listas de imágenes de referencia para el candidato de fusión temporal derivado de TMVP siempre se establece en 0, mientras que para AMVP, se establece igual al índice de referencia descodificado. En HEVC, un SPS puede incluir un indicador (por ejemplo, `sps_temporal_mvp_enable_flag`) y la cabecera del fragmento puede incluir un indicador (por ejemplo, `pic_temporal_mvp_enable_flag`) cuando `sps_temporal_mvp_enable_flag` es igual a 1. Cuando tanto `pic_temporal_mvp_enable_flag` como un id temporal son iguales a 0 para una imagen particular, no se usa ningún vector de movimiento de imágenes antes de esa imagen particular en el orden de descodificación como un TMVP en la descodificación de la imagen particular o una imagen después de la imagen particular en el orden de descodificación.

[0066] En la siguiente sección, se analizarán las técnicas de codificación multivista (por ejemplo, como en H.264/MVC) y multivista más profundidad (por ejemplo, como en 3D-HEVC). Inicialmente, se analizarán las técnicas de MVC. Como se indica anteriormente, MVC es una extensión de codificación multivista de ITU-T H.264/AVC. En MVC, los datos para una pluralidad de vistas se codifican en orden de tiempo primero, y en consecuencia, la disposición de orden de descodificación se denomina codificación de tiempo primero. En particular, se pueden codificar componentes de vista (es decir, imágenes) para cada una de la pluralidad de vistas en una instancia de tiempo común, a continuación se puede codificar otro conjunto de componentes de vista para una instancia de tiempo diferente y así sucesivamente. Una unidad de acceso puede incluir imágenes codificadas de todas las vistas para una instancia de tiempo de salida. Se debe entender que el orden de descodificación de las unidades de acceso no es necesariamente idéntico al orden de salida (o visualización).

[0067] Una orden de descodificación de MVC típica (es decir, orden de flujo de bits) se muestra en la FIG. 2. La disposición del orden de descodificación se denomina codificación de tiempo primero. Cabe destacar que el orden de descodificación de las unidades de acceso puede no ser idéntico al orden de salida o visualización. En la FIG. 2, S0-S7 se refieren, cada uno, a diferentes vistas del vídeo multivista. T0-T8 representa, cada una, 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-S7 para la instancia de tiempo T0, una segunda unidad de acceso puede incluir todas las vistas S0-S7 para la instancia de tiempo T1, etc.

[0068] Para propósitos de brevedad, la divulgación puede usar las siguientes definiciones:

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

[0069] En la FIG. 2, cada una de las vistas incluye conjuntos de imágenes. Por ejemplo, la vista S0 incluye el conjunto de imágenes 0, 8, 16, 24, 32, 40, 48, 56 y 64, la vista S1 incluye el conjunto de imágenes 1, 9, 17, 25, 33, 41, 49, 57 y 65, etc. Para la codificación de vídeo 3D, por ejemplo, 3D-HEVC, cada imagen puede incluir dos imágenes componentes: una imagen componente se denomina componente de vista de textura y la otra imagen componente se denomina componente de vista de profundidad. El componente de 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 la presente 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 siendo parte el componente de vista de profundidad de una misma vista de una única unidad de acceso.

[0070] El componente de vista de textura incluye el contenido de la imagen real que se visualiza. Por ejemplo, el componente de vista de textura puede incluir los componentes de luma (Y) y croma (Cb y Cr). El componente de vista de profundidad puede indicar profundidades relativas de los píxeles en su componente de vista de textura correspondiente. Como un ejemplo, el componente de vista de profundidad es 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 medida de las profundidades relativas de los píxeles en el componente de vista de textura.

[0071] Por ejemplo, un píxel blanco puro en el componente de vista de profundidad indica que su píxel o píxeles correspondientes en el componente de vista de textura correspondiente está más cercano de la perspectiva del observador, y un píxel negro puro en el componente de vista de profundidad indica que su píxel o píxeles correspondientes en el componente de vista de textura correspondiente está más alejado de 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 solo es necesaria la escala de grises para identificar la profundidad de 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.

[0072] El componente de vista de profundidad que usa solo valores de luma (por ejemplo, valores de intensidad) para identificar la profundidad se proporciona para propósitos ilustrativos y no se debe considerar limitante. En otros ejemplos, se puede utilizar cualquier técnica para indicar las profundidades relativas de los píxeles en el componente de vista de textura.

[0073] En la FIG. 3 se muestra una estructura de predicción de MVC típica (que incluye tanto la predicción entre imágenes dentro de cada vista como la predicción entre vistas) para la codificación de vídeo multivista. Las direcciones de predicción se indican por flechas, usando, el objeto al que se apunta, el objeto desde el que se apunta como referencia de predicción. En MVC, la predicción entre vistas se admite por la compensación de movimiento por disparidad, que usa la sintaxis de la compensación de movimiento de la H.264/AVC, pero permite que se use una imagen en una vista diferente como imagen de referencia.

[0074] En el ejemplo de la FIG. 3, se ilustran ocho vistas (que tienen las ID de vista de "S0" hasta "S7"), y se ilustran doce ubicaciones temporales (de "T0" hasta "T11") para cada vista. Es decir, cada fila en la FIG. 3 corresponde a una vista, mientras que cada columna indica una ubicación temporal.

[0075] Aunque la MVC tiene una denominada vista básica que se puede descodificar por los descodificadores de la H.264/AVC y se podrían admitir también los pares de vistas en estéreo por la MVC, la ventaja de MVC es que podría admitir un ejemplo que usa más de dos vistas como una entrada de vídeo 3D y que descodifica este vídeo 3D representado por las múltiples vistas. Un presentador de un cliente que tiene un descodificador de MVC puede esperar contenido de vídeo en 3D con múltiples vistas.

[0076] Las imágenes en la FIG. 3 se indican en la intersección de cada fila y cada columna. La norma H.264/AVC puede usar el término trama para representar una porción del vídeo. La presente divulgación puede usar el término imagen y trama de manera intercambiable.

[0077] Las imágenes en la FIG. 3 se ilustran usando un bloque que incluye una letra, designando la letra si la imagen correspondiente está intracodificada (es decir, una imagen I), o intercodificada en una dirección (es decir, como una imagen P) o en múltiples direcciones (es decir, como una imagen B). En general, las predicciones se indican por 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 desde la imagen I de la vista S0 en la ubicación temporal T0.

[0078] Al igual que con la codificación de vídeo de vista única, las imágenes de una secuencia de vídeo de codificación de vídeo multivista se pueden codificar predictivamente con respecto a las imágenes en diferentes ubicaciones temporales. Por ejemplo, la imagen b de la vista S0 en la ubicación temporal T1 tiene una flecha apuntando a la misma desde la imagen I de la vista S0 en la ubicación temporal T0, indicando que la imagen b se predice desde la imagen I. Adicionalmente, sin embargo, en el contexto de la codificación de vídeo multivista, 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 MVC, por ejemplo, la predicción entre vistas se realiza como si el componente de vista de otra vista fuera una referencia de predicción inter. Las potenciales referencias entre vistas se señalan en la extensión de MVC del conjunto de parámetros de secuencia (SPS) y se pueden modificar por el procedimiento de construcción de la lista de imágenes de referencia, que habilita el ordenamiento flexible de las referencias de predicción inter o predicción entre vistas. La predicción entre vistas también es un rasgo característico de una extensión multivista propuesta de la HEVC, que incluye la 3D-HEVC (multivista más profundidad).

[0079] La FIG. 3 proporciona diversos ejemplos de predicción entre vistas. Las imágenes de la vista S1, en el ejemplo de la FIG. 3, se ilustran como que se predicen a partir de imágenes en diferentes ubicaciones temporales de la vista S1, así como que se predicen 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.

[0080] En algunos ejemplos, la FIG. 3 se puede ver como una ilustración de los componentes de vista de textura. Por ejemplo, las imágenes I, P, B y b, ilustradas en la FIG. 2, se pueden considerar componentes de vista de textura para cada una de las vistas. De acuerdo con las técnicas descritas en la presente divulgación, para cada uno de los componentes de vista de textura ilustrados en la FIG. 3 existe un componente de vista de profundidad correspondiente. En algunos ejemplos, los componentes de vista de profundidad se pueden predecir de una manera similar a la ilustrada en la FIG. 3 para los componentes de vista de textura correspondientes.

[0081] La codificación de dos vistas también se podría admitir en MVC. Una de las ventajas de la MVC es que un codificador de MVC podría tomar más de dos vistas como una entrada de vídeo 3D y un descodificador de MVC puede descodificar una representación multivista de este tipo. Como tal, cualquier presentador con un descodificador de MVC puede esperar contenidos de vídeo 3D con más de dos vistas.

[0082] En 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 esta está en una vista diferente, pero dentro de la misma instancia de tiempo. Una imagen de referencia entre vistas se puede colocar en cualquier posición de una lista de imágenes de referencia, tal como cualquier imagen de referencia de predicción inter. Como se muestra en la FIG. 3, un componente de vista puede usar los componentes de vista en otras vistas como referencia. En MVC, la predicción entre vistas se realiza como si el componente de vista en otra vista fuera una referencia de predicción inter.

[0083] En el contexto de codificación de vídeo multivista, en general, existen dos tipos de vectores de movimiento. Uno se denomina un vector de movimiento normal. El vector de movimiento normal apunta a imágenes de referencia temporales y la predicción inter temporal correspondiente es predicción compensada por movimiento (MCP). El otro vector de movimiento es un vector de movimiento de disparidad (DMV). El DMV apunta a imágenes en una vista diferente (es decir, imágenes de referencia entre vistas) y la predicción inter correspondiente es predicción compensada por disparidad (DCP).

[0084] Otro tipo de formato de codificación de vídeo multivista introduce el uso de valores de profundidad (por ejemplo, como en 3D-HEVC). Para el formato de datos de vídeo multivista más profundidad (MVD), que es popular para la televisión 3D y los videos de puntos de vista gratuitos, las imágenes de textura y los mapas de profundidad se pueden codificar con imágenes de textura multivista de forma independiente. La FIG. 4 ilustra el formato de datos MVD con una imagen de textura y su mapa de profundidad por muestra asociado. El rango de profundidad se puede restringir para estar en el rango de distancia $Z_{cercana}$ mínima y $Z_{lejána}$ máxima de la cámara para los puntos 3D correspondientes.

[0085] Los parámetros de cámara y los valores de rango de profundidad pueden ser útiles para procesar componentes de vista descodificados antes de su representación en un visualizador 3D. Por lo tanto, se define un mensaje de información de mejora complementaria (SEI) especial para la versión actual de H.264/MVC, es decir, SEI de información de adquisición multivista, que incluye información que especifica diversos parámetros del entorno de adquisición. Sin embargo, no existen sintaxis especificadas en H.264/MVC para indicar la información relacionada con el rango de profundidad.

[0086] Ahora se analizará la particionamiento de movimiento asimétrica (AMP) y los tamaños de bloque de compensación de movimiento en HEVC. En HEVC, los bloques de codificación intercodificados se pueden particionar en una, dos o cuatro separaciones. Son posibles diversas conformaciones de dichas separaciones. Las posibilidades de particionamiento de ejemplo para bloques de codificación con predicción inter se representan en la FIG. 5.

[0087] La fila superior de las separaciones en la FIG. 5 ilustra las denominadas separaciones simétricas. Una particionamiento $N \times N$ es simplemente un bloque de codificación que no se ha separado. Una particionamiento $N/2 \times N$ es un bloque de codificación separado en dos separaciones rectangulares verticales. Del mismo modo, una particionamiento $N \times N/2$ es un bloque de codificación separado en dos separaciones rectangulares horizontales. Una particionamiento $N/2 \times N/2$ es un bloque de codificación separado en cuatro separaciones cuadradas iguales.

[0088] Los cuatro tipos de particionamiento inferiores en la FIG. 5 se denominan separaciones asimétricas y se pueden usar en particionamiento de movimiento asimétrica (AMP) para predicción inter. Una particionamiento del modo AMP tiene la altura o el ancho $N/4$ y el ancho o la altura N , respectivamente, y la otra particionamiento consiste en el resto del CB al tener una altura o ancho de $3N/4$ y ancho o altura N . A cada particionamiento intercodificada se le asignan uno o dos vectores de movimiento e índices de imagen de referencia (es decir, un vector de movimiento e índice de referencia para la predicción unidireccional y dos vectores de movimiento e índices de referencia para la predicción bidireccional). En algunos ejemplos, para minimizar el ancho de banda de memoria en el peor de los casos, las separaciones de tamaño 4×4 no están permitidas para la predicción inter y las separaciones de tamaños 4×8 y 8×4 están restringidas a la codificación unipredictiva que se basa en una lista de datos predictivos.

[0089] Como se analizará con más detalle a continuación, la presente divulgación describe las técnicas para AMP cuando se usan junto con técnicas de codificación 3D-HEVC, incluyendo la predicción de síntesis de vista inversa (BVSP).

[0090] Lo siguiente describe la lista de candidatos de fusión en HEVC. Por ejemplo, la lista de candidatos de fusión se puede construir con las siguientes etapas. Para candidatos de fusión espaciales, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden derivar hasta cuatro candidatos de vector de movimiento espaciales a partir de cinco bloques contiguos espaciales, como se ilustra en la FIG. 6.

[0091] El orden en el que el codificador de vídeo 20 y el descodificador de vídeo 30 pueden evaluar los bloques contiguos espaciales es como sigue: izquierda (A1), superior (B1), superior derecha (B0), inferior izquierda (A0) y superior izquierda (B2), como se muestra en la FIG. 6. En algunos ejemplos, se puede aplicar un procedimiento de depuración para eliminar candidatos de vector de movimiento que tienen información de movimiento idéntica (por ejemplo, vectores de movimiento e índices de referencia). Por ejemplo, los vectores de movimiento y los índices de referencia de B1 se pueden comparar con los vectores de movimiento y los índices de referencia de A1, los vectores de movimiento y los índices de referencia de B0 se pueden comparar con los vectores de movimiento y los índices de referencia de B1, los vectores de movimiento y los índices de referencia de A0 se pueden comparar con los vectores de movimiento y los índices de referencia de A1, y los vectores de movimiento y los índices de referencia de B2 se pueden comparar con los vectores de movimiento y los índices de referencia tanto de B1 como de A1. A continuación, uno de los dos candidatos que tiene información de movimiento idéntica se puede eliminar de la lista de candidatos de vector de movimiento. Si ya hay cuatro candidatos disponibles después del procedimiento de depuración, el candidato B2 no se inserta en la lista de candidatos de fusión.

[0092] Un candidato del predictor de vector de movimiento temporal (TMVP) coubicado de una imagen de referencia, si está habilitado y disponible, se añade en la lista de candidatos de vector de movimiento después de los candidatos de vector de movimiento espaciales.

[0093] Si una lista de candidatos de vector de movimiento no está completa (por ejemplo, tiene menos de un número predeterminado de entradas, se pueden generar e insertar uno o más candidatos de vector de movimiento artificiales al final de la lista de candidatos de fusión. Los tipos de candidatos de vector de movimiento artificiales de ejemplo incluyen candidatos de fusión bipredictivos combinados derivados solo para fragmentos B, y candidatos de fusión de vector de movimiento cero si no hay suficientes candidatos de fusión bipredictivos (u otro tipo de candidatos de vector de movimiento artificiales) para proporcionar el número predeterminado de candidatos de vector de movimiento.

[0094] Cuando un fragmento actual es un fragmento B, se invoca el procedimiento de derivación de candidatos de fusión bipredictivos combinados. Para cada par de candidatos que ya están en la lista de candidatos y tienen la información de movimiento necesaria, los candidatos de vector de movimiento bipredictivos combinados (con un índice indicado por combldx) se derivan usando una combinación del vector de movimiento del primer candidato (con índice

de candidato de fusión igual a $l0CandIdx$ que se refiere a una imagen en la lista 0 (si está disponible), y el vector de movimiento de un segundo candidato (con índice de candidato de fusión igual a $l1CandIdx$ que se refiere a una imagen en la lista 1 (si está disponible y la imagen de referencia o bien el vector de movimiento es diferente del primer candidato).

5 **[0095]** La FIG. 7 es una tabla que indica una especificación de ejemplo de $l0CandIdx$ e $l1CandIdx$ en 3D-HEVC. Por ejemplo, la FIG. 7 ilustra las definiciones de $l0CandIdx$ e $l1CandIdx$ correspondientes a $combIdx$.

10 **[0096]** Para $combIdx$ siendo 0... 11, el procedimiento de generación de candidatos de vector de movimiento bipredictivos combinados finaliza cuando se cumple una de las siguientes condiciones: (1) $combIdx$ es igual a $(numOrigMergeCand * (numOrigMergeCand - 1))$ en el que $numOrigMergeCand$ indica el número de candidatos en la lista de fusión antes de invocar este procedimiento; (2) el número total de candidatos (incluyendo los candidatos de fusión bipredictivos combinados recientemente generados) en la lista de fusión es igual a $MaxNumMergeCand$.

15 **[0097]** Esta sección describe la derivación de candidatos de fusión de vector de movimiento cero. Para cada candidato, los vectores de movimiento cero y un índice de imagen de referencia se establecen de 0 al número de índice de imagen de referencia disponible menos 1. Si todavía hay menos candidatos que el número máximo de candidatos de vector de movimiento de fusión (por ejemplo, como se indica por el elemento sintáctico $MaxNumMergeCand$), se inserta un índice de referencia cero y un vector de movimiento hasta que el número total de candidatos sea igual a $MaxNumMergeCand$.

20

[0098] Lo siguiente describe la restricción de tamaño de compensación de movimiento en HEVC. Para minimizar el ancho de banda de memoria en el peor de los casos, las separaciones de tamaño 4x4 no están permitidas para la predicción inter y las separaciones de tamaños 4x8 y 8x4 están restringidas a la codificación unipredictiva.

25

[0099] Para satisfacer una restricción de este tipo mencionada anteriormente, cuando el tamaño de PU actual es igual a 8x4 o 4x8, el candidato de fusión bipredictivo espacial/temporal/combinado generado, si está asociado con el modo de bipredicción, la PU actual se debe restablecer para usar unipredicción modificando la dirección de predicción a lista 0, y el índice de imagen de referencia y el vector de movimiento correspondiente a $RefPicList1$ a -1 y (0, 0), respectivamente.

30

[0100] Como se menciona anteriormente, una 3D-HEVC está en fase de desarrollo. 3D-HEVC puede mejorar la eficacia de codificación usando predicción de movimiento entre vistas y predicción residual entre vistas. En otras palabras, para mejorar además la eficiencia de codificación, se han adoptado dos nuevas tecnologías, a saber, "predicción de movimiento entre vistas" y "predicción residual entre vistas" en el software de referencia. En la predicción de movimiento entre vistas, un codificador de vídeo (por ejemplo, el codificador de vídeo 20 o el descodificador de vídeo 30) puede determinar (es decir, predecir) la información de movimiento de una PU actual en base a la información de movimiento de una PU en una vista diferente de la PU actual. En la predicción residual entre vistas, un codificador de vídeo puede determinar bloques residuales de una CU actual en base a datos residuales en una vista diferente de la CU actual.

35

40

[0101] Ahora se analizará la derivación del vector de disparidad basado en bloques contiguos (NBDV) en 3D-HEVC. La derivación de NBDV se usa como una técnica de derivación de vector de disparidad en 3D-HEVC debido al hecho de que 3D-HEVC usa el orden de codificación de textura primero para todas las vistas. Puesto que el mapa de profundidad correspondiente no está disponible para una imagen de textura actualmente codificada, en NBDV, se deriva un vector de disparidad de los bloques contiguos. En algunas de las propuestas para el diseño de 3D-HEVC, el vector de disparidad derivado del procedimiento de derivación de NBDV se podría refinar además recuperando los datos de profundidad correspondientes a una vista de textura de referencia.

45

50 **[0102]** 3D-HEVC adoptó inicialmente las técnicas de derivación de NBDV propuestas en JCT3V-A0097 (3D-CE5.h: Disparity vector generation results [3D-CE5.h: resultados de la generación del vector de disparidad], L. Zhang, Y. Chen, M. Karczewicz (Qualcomm)). Se incluyeron vectores de disparidad implícita con un NBDV simplificado en JCTVC-A0126 (3D-CE5.h: Simplification of disparity vector derivation for HEVC-based 3D video coding [3D-CE5.h: simplificación de la derivación del vector de disparidad para codificación de vídeo 3D basada en HEVC], J. Sung, M. Koo, S. Yea (LG)). Además, en JCT3V-B0047 (3D-CE5.h related: Improvements for disparity vector derivation [3D-CE5.h relacionada: mejoras para la derivación del vector de disparidad], J. Kang, Y. Chen, L. Zhang, M. Karczewicz (Qualcomm)), las técnicas de derivación de NBDV se simplificaron adicionalmente al eliminar los vectores de disparidad implícitos almacenados en la memoria intermedia de imágenes descodificadas, pero también mejoraron una ganancia de codificación con la selección de imágenes de acceso aleatorio (RAP). Se describieron técnicas adicionales para la derivación de NBDV en JCT3V-D0181 (CE2: CU-based Disparity Vector Derivation in 3D-HEVC [CE2: derivación del vector de disparidad basado en CU en 3D-HEVC], J. Kang, Y. Chen, L. Zhang, M. Karczewicz (Qualcomm)).

55

60

[0103] Se usa un vector de disparidad (DV) como estimador del desplazamiento entre dos vistas. Debido a que los bloques contiguos comparten casi la misma información de movimiento/disparidad en la codificación de vídeo, el bloque actual puede usar la información de vectores de movimiento de bloques contiguos como un buen predictor.

65

Siguiendo esta idea, el procedimiento de derivación de NBDV usa la información de disparidad contigua para estimar el vector de disparidad en diferentes vistas.

[0104] Para implementar NBDV, el codificador de vídeo 20 define inicialmente varios bloques contiguos espaciales y temporales. A continuación, el codificador de vídeo 20 verifica cada uno de los bloques contiguos en un orden predefinido determinado por la prioridad de la correlación entre el bloque actual y el bloque candidato. Una vez que se encuentra un vector de movimiento de disparidad (es decir, el vector de movimiento apunta a una imagen de referencia entre vistas) en los candidatos, el codificador de vídeo 20 convierte el vector de movimiento de disparidad en un vector de disparidad y también se devuelve el índice de orden de vista asociado. Se utilizan dos conjuntos de bloques contiguos. Un conjunto incluye bloques contiguos espaciales y el otro conjunto incluye bloques contiguos temporales.

[0105] En propuestas recientes para 3D-HEVC, se usan dos bloques contiguos espaciales en la derivación de NBDV. Los bloques contiguos espaciales son los bloques contiguos izquierdos y superiores con respecto a la unidad de codificación (CU) actual 90, como se indica por A1 y B1, respectivamente, en la FIG. 8. Cabe destacar que los bloques contiguos representados en la FIG. 8 están en la misma ubicación que algunos de los bloques contiguos usados en el modo de FUSIÓN en HEVC. Por lo tanto, no se requiere acceso adicional a la memoria. Sin embargo, se debe entender que también se pueden usar bloques contiguos en otras ubicaciones con respecto a la CU 90 actual.

[0106] Para verificar bloques contiguos temporales, el codificador de vídeo 20 primero realiza un procedimiento de construcción para una lista de imágenes candidatas. Hasta dos imágenes de referencia de una vista actual se pueden tratar como imágenes candidatas. El codificador de vídeo 20 primero añade una imagen de referencia coubicada a la lista de imágenes candidatas, seguida del resto de imágenes candidatas en el 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 imagen de referencia en la misma lista de imágenes de referencia que la imagen coubicada precede a la otra imagen de referencia que tiene el mismo índice de referencia. Para cada imagen candidata en la lista de imágenes candidatas, el codificador de vídeo 20 determina el bloque de la región coubicada que abarca la posición central como el bloque contiguo temporal.

[0107] Cuando se codifica un bloque con predicción de movimiento entre vistas, se puede derivar un vector de disparidad para seleccionar un bloque correspondiente en una vista diferente. El vector de disparidad derivado en el procedimiento de predicción de movimiento entre vistas se denomina vector de disparidad implícita (IDV o también conocido como vector de disparidad derivado). Aunque el bloque esté codificado con predicción de movimiento, el vector de disparidad derivado no se descarta para el propósito de codificar un bloque siguiente.

[0108] En un diseño del HTM, durante el procedimiento de derivación de NBDV, un codificador de vídeo (por ejemplo, el codificador de vídeo 20 y/o el decodificador de vídeo 30) se configura para verificar los vectores de movimiento de disparidad en los bloques contiguos temporales, los vectores de movimiento de disparidad en los bloques contiguos espaciales y a continuación los IDV, en orden. Una vez que se encuentra el vector de movimiento de disparidad, o IDV, finaliza el procedimiento.

[0109] Ahora se analizará el ajuste del procedimiento de derivación de NBDV (NBDV-R) al acceder a la información de profundidad. Cuando un vector de disparidad se deriva del procedimiento de derivación de NBDV, el vector de disparidad derivado se puede refinar además recuperando los datos de profundidad del mapa de profundidad de la vista de referencia. El procedimiento de ajuste puede incluir las siguientes técnicas:

a) Ubicar un bloque de profundidad correspondiente por el vector de disparidad derivado en la vista de profundidad de referencia previamente codificada, tal como la vista básica; el tamaño del bloque de profundidad correspondiente es el mismo que el de la PU actual.

b) Seleccionar un valor de profundidad de cuatro píxeles de esquina del bloque de profundidad correspondiente y convertir el valor de profundidad en el componente horizontal del vector de disparidad refinado. El componente vertical del vector de disparidad no se modifica.

[0110] Cabe destacar que, en algunos ejemplos, el vector de disparidad refinado se puede usar para la predicción de movimiento entre vistas, mientras que el vector de disparidad no refinado se puede usar para la predicción residual entre vistas. Además, el vector de disparidad refinado se puede almacenar como el vector de movimiento de una PU si se codifica la PU con el modo de predicción de síntesis de vista inversa. En algunas propuestas para 3D-HEVC, se puede acceder al componente de vista de profundidad de la vista básica independientemente del valor del índice de orden de vista derivado del procedimiento de derivación de NBDV.

[0111] Ahora se analizarán las técnicas de predicción de síntesis de vista inversa (BVSP) en 3D-HEVC. Un ejemplo de enfoque BVSP, como se propone por D. Tian, *et al.* "CE1.h: Backward View Synthesis Prediction Using Neighboring Blocks", JCT3V-C0152, disponible en http://phenix.it-sudparis.eu/jct2/doc_end_user/current_document.php?id=594_, se adoptó en la 3.^a conferencia de JCT-3V. La idea básica de BVSP es similar a la predicción de síntesis de vista basada en bloques en 3D-AVC. Ambas técnicas usan distorsión inversas y predicción de síntesis de vista basada en

bloques para evitar transmitir las diferencias de vectores de movimiento y usan vectores de movimiento más precisos. Los detalles de implementación de BVSP en 3D-HEVC y 3D-AVC son diferentes debido a las diferentes plataformas.

5 **[0112]** En 3D-HEVC, el modo BVSP se admite para un bloque intercodificado que se codifica en modo de salto o bien de fusión. En una propuesta de ejemplo para 3D-HEVC, el modo BVSP no está permitido para un bloque codificado en el modo de predicción de vectores de movimiento avanzada (AMVP). En lugar de transmitir un indicador para indicar el uso del modo BVSP, el codificador de vídeo 20 se puede configurar para añadir un candidato de fusión adicional (es decir, candidato de fusión BVSP) a la lista de candidatos de fusión, y cada candidato se asocia con un indicador de BVSP. Cuando el índice de fusión descodificado corresponde a un candidato de fusión BVSP, el índice de fusión descodificado indica que la unidad de predicción (PU) actual usa el modo BVSP. Para cada subbloque dentro de la PU actual, se puede derivar un vector de movimiento de disparidad al convertir un valor de profundidad en una vista de referencia de profundidad.

15 **[0113]** El establecimiento de los indicadores de BVSP se define como sigue. Cuando un bloque contiguo espacial usado para derivar un candidato de fusión espacial se codifica con el modo BVSP, la información de movimiento asociada del candidato de fusión espacial se hereda por el bloque actual, como en el modo de fusión convencional. Además, este candidato de fusión espacial está asociado con un indicador de BVSP igual a 1 (es decir, que indica que el candidato de fusión espacial se codificó con el modo BVSP). Para el candidato de fusión BVSP recién presentado, el indicador de BVSP se establece en 1. Para todos los demás candidatos de fusión, los indicadores de BVSP asociados se establecen en 0.

25 **[0114]** Como se analiza anteriormente, en 3D-HEVC, el codificador de vídeo 20 se puede configurar para derivar e insertar un nuevo candidato, denominado candidato de fusión BVSP, en la lista de candidatos de fusión. Los índices de imagen de referencia y los vectores de movimiento correspondientes se establecen por el siguiente procedimiento.

30 **[0115]** El primer codificador de vídeo 20 se puede configurar para obtener el índice de vista indicado por el elemento sintáctico de índice de vista (por ejemplo, refVIdxLX en 3D-HEVC) del vector de disparidad derivado del procedimiento de derivación de NBDV. El codificador de vídeo 20 también se puede configurar para obtener la lista de imágenes de referencia (por ejemplo, RefPicListX (RefPicList0 o bien RefPicList1)) que está asociada con la imagen de referencia con el índice de orden de vista igual a refVIdxLX. A continuación, el codificador de vídeo 20 usa el índice de imagen de referencia correspondiente y el vector de disparidad obtenido del procedimiento de derivación de NBDV como información de movimiento del candidato de fusión BVSP en RefPicListX (es decir, RefPicList0 o bien RefPicList1).

35 **[0116]** Si el fragmento actual es un fragmento B, el codificador de vídeo 20 verifica la disponibilidad de una imagen de referencia entre vistas con un índice de orden de vista indicado por refVIdxLY diferente a refVIdxLX en la lista de imágenes de referencia distinta de RefPicListX, es decir, RefPicListY siendo Y 1-X. Si se encuentra una imagen de referencia entre vistas diferente de este tipo, el codificador de vídeo 20 realiza la predicción de síntesis de vista bipredictiva. El codificador de vídeo 20 se puede configurar además para usar el índice de imagen de referencia correspondiente de la imagen de referencia entre vistas diferente y el vector de disparidad ajustado a escala del procedimiento de derivación de NBDV como información de movimiento del candidato de fusión BVSP en RefPicListY. El bloque de profundidad de la vista que tiene un índice de orden de vista igual a refVIdxLX se usa como información de profundidad del bloque actual (en caso de orden de codificación de textura primero). El codificador de vídeo 20 sintetiza las dos imágenes de referencia entre vistas diferentes (una de cada lista de imágenes de referencia) por medio de un procedimiento de distorsión inversa y pondera además las imágenes de referencia sintetizadas para lograr el predictor de BVSP final.

45 **[0117]** Para tipos de fragmento distintos de un fragmento B (por ejemplo, un fragmento P), el codificador de vídeo 20 aplica la predicción de síntesis de vista unipredictiva con RefPicListX como lista de imágenes de referencia para la predicción.

50 **[0118]** En 3D-HTM, la codificación de textura primero se aplica en condiciones de prueba comunes. Dado que el componente de textura de una vista se codifica antes que el componente de profundidad, el componente de profundidad no básico correspondiente no está disponible cuando se descodifica un componente de textura no básico. Por lo tanto, el descodificador de vídeo 30 se puede configurar para estimar la información de profundidad, y a continuación usar la información de profundidad estimada para realizar BVSP. Para estimar la información de profundidad para un bloque, se propone derivar primero un vector de disparidad de los bloques contiguos (por ejemplo, usando el procedimiento de derivación de NBDV), y a continuación usar el vector de disparidad derivado para obtener un bloque de profundidad de una vista de referencia.

55 **[0119]** La FIG. 9 ilustra técnicas de ejemplo para ubicar un bloque de profundidad desde la vista de referencia, y a continuación usar el bloque de profundidad para la predicción de BVSP. Inicialmente, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden utilizar un vector de disparidad 104 asociado con el bloque contiguo 102. Es decir, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden acceder a la información de vector de disparidad de bloques contiguos ya codificados (tal como el bloque contiguo 102) y reutilizar cualquier información de vector de disparidad asociada para el bloque actual 100. El vector de disparidad 104 apunta al bloque de profundidad 106 en una imagen de profundidad de referencia. Cuando el vector de disparidad 104 se reutiliza para el bloque actual 100,

el vector de disparidad 104 ahora apunta al bloque de profundidad 108 en la imagen de profundidad de referencia. El bloque de profundidad 108 corresponde al bloque actual 100. A continuación, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden usar la información de profundidad en la imagen de profundidad de referencia 108 para sintetizar un bloque en una imagen de textura de referencia usando técnicas de distorsión inversa. A continuación, se puede usar la imagen de textura sintetizada como una imagen de referencia para predecir el bloque actual 100.

[0120] En un ejemplo de la presente divulgación, para el procedimiento de derivación de NBDV, permítase que (dv_x, dv_y) indique el vector de disparidad 104 identificado por el procedimiento de derivación de NBDV, e indique la posición del bloque actual 100 como $(bloque_x, bloque_y)$. En un ejemplo de BVSP unipredictiva, el codificador de vídeo 20 y/o el descodificador de vídeo 30 se pueden configurar para capturar un bloque de profundidad 108 que tiene una posición superior izquierda de $(bloque_x + dv_x, bloque_y + dv_y)$ en el componente de vista de profundidad de la vista de referencia. El codificador de vídeo 20 y/o el descodificador de vídeo 30 se pueden configurar para particionar primero el bloque actual 100 (por ejemplo, una PU) en varios subbloques, cada uno con el mismo tamaño (por ejemplo, igual a $W \times H$). Para cada subbloque con un tamaño igual a $W \times H$, el codificador de vídeo 20 y/o el descodificador de vídeo 30 identifica el valor de profundidad máxima de los cuatro píxeles de esquina del subbloque de profundidad 108 correspondiente dentro del componente de vista de profundidad capturado, por ejemplo, como se muestra en la FIG. 10. La FIG. 10 es un diagrama conceptual que ilustra cuatro píxeles de esquina de un bloque de profundidad 8×8 110. Los cuatro píxeles de esquina se pueden marcar como un píxel superior izquierdo (TL), un píxel superior derecho (TR), un píxel inferior izquierdo (BL) y un píxel inferior derecho (BR). El codificador de vídeo 20 y/o el descodificador de vídeo 30 convierte el valor de profundidad máxima en un vector de movimiento de disparidad. A continuación, el vector de movimiento de disparidad derivado para cada subbloque se usa para la compensación de movimiento.

[0121] Esta sección analizará BVSP cuando se realiza predicción bidireccional. Cuando hay múltiples imágenes de referencia entre vistas de diferentes vistas en RefPicList0 y RefPicList1, el codificador de vídeo 20 y/o el descodificador de vídeo 30 aplica BVSP bipredictiva. En BVSP de bipredicción, se generarán dos predictores de predicción de síntesis de vista (es decir, dos bloques de referencia sintetizados) de cada lista de referencia, como se describe anteriormente. A continuación, los dos predictores de predicción de síntesis de vista se promedian para obtener el predictor de predicción de síntesis de vista final.

[0122] El tamaño de compensación de movimiento, es decir, $W \times H$ como se describe anteriormente, podría ser 8×4 o bien 4×8 . En un ejemplo, para determinar el tamaño de compensación de movimiento, se aplica la siguiente regla:

Para cada bloque 8×8 , se verifican cuatro esquinas del bloque 8×8 de profundidad correspondiente y:

$$si (vdepth[TL] < vdepth[BR]?0:1) != (vdepth[TR] < vdepth[BL]?0:1)$$

usar separación 4×8 ($W = 4, H = 8$)

si no

usar separación 8×4 ($W = 8, H = 4$)

[0123] Lo siguiente describe el procedimiento de derivación de candidatos entre vistas para el modo de salto/fusión en una propuesta para 3D-HEVC. En base al vector de disparidad derivado del procedimiento de derivación de NBDV, un nuevo candidato de vector de movimiento, llamado candidato de vector de movimiento predicho entre vistas (IPMVC), si está disponible, se puede añadir a AMVP y a la lista de candidatos de vector de movimiento de modo de salto/fusión. El vector de movimiento predicho entre vistas, si está disponible, es un vector de movimiento temporal. Puesto que los modos de salto tienen el mismo procedimiento de derivación del vector de movimiento que el modo de fusión, todas las técnicas descritas en el presente documento se aplican a ambos modos de fusión y salto.

[0124] La FIG. 11 es un diagrama conceptual que ilustra una derivación de ejemplo del candidato de vector de movimiento predicho entre vistas para el modo de fusión/salto. Por ejemplo, la FIG. 11 muestra un ejemplo del procedimiento de derivación del candidato de vector de movimiento predicho entre vistas. Para el modo de fusión/salto, el candidato de vector de movimiento predicho entre vistas se deriva por las siguientes etapas. Primero, el codificador de vídeo 20 y/o el descodificador de vídeo 30 ubica, usando el vector de disparidad, un bloque correspondiente (por ejemplo, un bloque de referencia) 112 de la PU/CU actual 114 en una vista de referencia de la misma unidad de acceso. En el ejemplo de la FIG. 11, el bloque actual (PU actual) 114 está en la vista V1, mientras que el bloque de referencia 112 correspondiente está a la vista V0. Si el bloque de referencia 112 correspondiente no está intracodificado y no está predicho entre vistas, y su imagen de referencia (en este ejemplo en la vista V0 y el tiempo T1) tiene un valor POC igual al de una entrada en la misma lista de imágenes de referencia de la PU/CU actual 114, la información de movimiento (es decir, dirección de predicción, índice de imagen de referencia y vector de movimiento) del bloque de referencia 112 correspondiente se deriva para que sea el vector de movimiento predicho entre vistas después de convertir el índice de referencia en base al POC de la imagen de referencia.

[0125] El bloque de referencia 112 correspondiente se puede definir como sigue. Primero se indica una ubicación de luma (xP, yP) de la muestra de luma superior izquierda de la unidad de predicción actual con respecto a la muestra

de luma superior izquierda de la imagen actual. Las variables nPSW y nPSH indican el ancho y la altura de la unidad de predicción actual, respectivamente. El índice de orden de vista de referencia se marca como refViewIdx y el vector de disparidad como mvDisp. La ubicación de luma de la capa de referencia (xRef, yRef) se deriva por:

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

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

[0126] El bloque de referencia 112 correspondiente se establece en la unidad de predicción que abarca la ubicación de luma (xRef, yRef) en el componente de vista con ViewIdx igual a refViewIdx.

[0127] Además, el vector de disparidad se puede convertir en un vector de movimiento de disparidad entre vistas (IDMVC), que se añade a la lista de candidatos de fusión en una posición diferente del IPMVC. El vector de movimiento de disparidad entre vistas también se puede añadir en la lista de candidatos de AMVP en la misma posición que IPMVC, cuando esté disponible. En este contexto, se puede denominar "candidato entre vistas" al IPMVC o bien al IDMVC.

[0128] En un ejemplo para el modo de fusión/salto, el IPMVC, si está disponible, se inserta antes de todos los candidatos de fusión espaciales y temporales en la lista de candidatos de fusión. El IDMVC se inserta antes del candidato de fusión espacial derivado de A₀.

[0129] La siguiente sección describe las construcciones de listas de candidatos de fusión para codificación de textura en 3D-HEVC. Primero, el codificador de vídeo 20 y/o el descodificador de vídeo 30 derivan un vector de disparidad, por ejemplo, usando las técnicas de derivación de NBDV descritas anteriormente. Después de derivar el vector de disparidad, el codificador de vídeo 20 y/o el descodificador de vídeo 30 se pueden configurar para ejecutar el procedimiento de construcción de la lista de candidatos de fusión en 3D-HEVC como se describe a continuación.

[0130] El codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden derivar uno o más IPMVC usando el procedimiento descrito anteriormente. Si hay un IPMVC disponible, el IPMV se puede insertar en la lista de fusión.

[0131] A continuación, el codificador de vídeo 20 y/o el descodificador de vídeo 30 se pueden configurar para derivar candidatos de fusión espaciales y una o más inserciones de IDMVC en 3D-HEVC. Para derivar candidatos de fusión espaciales, el codificador de vídeo 20 y/o el descodificador de vídeo 30 se pueden configurar para verificar la información de movimiento de las PU contiguas espaciales en el siguiente orden: A₁, B₁, B₀, A₀ o B₂, por ejemplo, como se muestra en la FIG. 6.

[0132] El codificador de vídeo 20 y/o el descodificador de vídeo 30 se pueden configurar además para realizar una depuración restringida. Para realizar una depuración restringida, el codificador de vídeo 20 y/o el descodificador de vídeo 30 se pueden configurar para no insertar el candidato de fusión espacial en la ubicación A₁ en la lista de candidatos de fusión si A₁ e IPMVC tienen los mismos vectores de movimiento y los mismos índices de referencia. De otro modo, el candidato de fusión espacial en la ubicación A₁ se inserta en la lista de candidatos de fusión.

[0133] Si el candidato de fusión en la ubicación B₁ y el candidato de fusión en la ubicación de fusión A₁ (o el IPMVC) tienen los mismos vectores de movimiento y los mismos índices de referencia, el candidato de fusión en la ubicación B₁ no se inserta en la lista de candidatos de fusión. De otro modo, el candidato de fusión en la ubicación B₁ se inserta en la lista de candidatos de fusión. Si el candidato de fusión en la ubicación B₀ está disponible (es decir, está codificado y tiene información de movimiento), el candidato de fusión en la ubicación B₀ se añade a la lista de candidatos. El codificador de vídeo 20 y/o el descodificador de vídeo 30 deriva el IDMVC usando el procedimiento descrito anteriormente. Si un IDMVC está disponible, y la información de movimiento del IDMVC es diferente de los candidatos derivados de A₁ y B₁, el IDMVC se inserta en la lista de candidatos.

[0134] Si la BVSP está habilitada para la imagen completa (o para el fragmento actual), a continuación se inserta el candidato de fusión de BVSP en la lista de candidatos de fusión. Si el candidato de fusión en la ubicación A₀ está disponible, se añade a la lista de candidatos. Si el candidato de fusión en B₂ está disponible, se añade a la lista de candidatos.

[0135] La siguiente sección analizará el procedimiento de derivación para el candidato de fusión temporal en 3D-HEVC. La derivación de candidatos de fusión temporales en 3D-HEVC es similar al procedimiento de derivación de candidatos de fusión temporales en HEVC, donde se utiliza la información de movimiento de la PU coubicada. Sin embargo, para 3D-HEVC, se puede cambiar el índice de imagen de referencia objetivo del candidato de fusión temporal en lugar de fijar el índice de imagen de referencia para que sea 0. Cuando un índice de referencia objetivo igual a 0 corresponde a una imagen de referencia temporal (en la misma vista), mientras que el vector de movimiento

de la unidad de predicción (PU) coubicada apunta a una imagen de referencia entre vistas, el índice de referencia objetivo cambia a otro índice que corresponde a la primera entrada de una imagen de referencia entre vistas en la lista de imágenes de referencia. Por el contrario, cuando el índice de referencia objetivo igual a 0 corresponde a una imagen de referencia entre vistas mientras que el vector de movimiento de la unidad de predicción (PU) coubicada apunta a una imagen de referencia temporal, el índice de imagen de referencia objetivo cambia a otro índice que corresponde a la primera entrada de una imagen de referencia temporal en la lista de imágenes de referencia.

[0136] Ahora se analizará un procedimiento de derivación de ejemplo para candidatos de fusión bipredictivos combinados en 3D-HEVC. Si el número total de candidatos derivados de las dos etapas anteriores (es decir, la derivación de candidatos de fusión espaciales y la derivación de candidatos de fusión temporales) es menor que el número máximo de candidatos (que puede estar predefinido), se realiza el mismo procedimiento como se define en HEVC, como se describe anteriormente. Sin embargo, la especificación de los índices de referencia $l0CandIdx$ e $l1CandIdx$ es diferente. La FIG. 12 es otra tabla que indica una especificación de ejemplo de $l0CandIdx$ e $l1CandIdx$ en 3D-HEVC. Por ejemplo, la relación entre $combIdx$, $l0CandIdx$ e $l1CandIdx$ se define en la tabla ilustrada en la FIG. 12.

[0137] Un procedimiento de derivación de ejemplo para candidatos de fusión de vector de movimiento cero en 3D-HEVC es el mismo procedimiento que se define en HEVC. En un ejemplo para 3D-HEVC, el número total de candidatos en la lista de candidatos de fusión es de hasta 6, y el elemento sintáctico `five_minus_max_num_merge_cand` se genera en la cabecera del fragmento para especificar el número máximo de candidatos de fusión restados a 6. Se debe notar que el valor del elemento sintáctico `five_minus_max_num_merge_cand` está en el intervalo de 0 a 5, inclusive.

[0138] Lo siguiente describe la herencia del vector de movimiento (MVI) para la codificación de profundidad, por ejemplo, en 3D-HEVC. Las técnicas de MVI buscan aprovechar la similitud de las características de movimiento entre los componentes de textura de una imagen y sus componentes de vista de profundidad asociados. La FIG. 13 es un diagrama conceptual que ilustra una derivación de ejemplo de un candidato heredado de vector de movimiento para codificación de profundidad. Por ejemplo, la FIG. 13 muestra un ejemplo del procedimiento de derivación del candidato de MVI donde el bloque de textura 120 correspondiente se selecciona como el bloque 4x4 ubicado en la parte inferior derecha del centro de la PU actual 122 en una imagen de textura 124. Para la PU actual 126 en la imagen de profundidad 128, el candidato de MVI se reutiliza usando los vectores de movimiento y los índices de referencia asociados con el bloque de textura 120 correspondiente ya codificado en la imagen de textura 124 correspondiente, si dicha información está disponible.

[0139] Cabe destacar que los vectores de movimiento con precisión de entero se usan en la codificación de profundidad, mientras que los vectores de movimiento con precisión de un cuarto se utilizan para la codificación de textura. Por lo tanto, el vector de movimiento del bloque de textura correspondiente se puede ajustar a escala antes de usarse como un candidato de MVI.

[0140] Con la generación de candidatos de MVI, la lista de candidatos de fusión para las vistas de profundidad se construye como sigue. Para la inserción de MVI, la MVI se deriva usando las técnicas como se describe anteriormente y, si está disponible, se inserta en la lista de candidatos de fusión.

[0141] El procedimiento de derivación para los candidatos de intercalación espaciales y la inserción de IDMVC en 3D-HEVC para la codificación de profundidad se describe a continuación. El primer codificador de vídeo 20 y/o decodificador de vídeo 30 se puede configurar para verificar la información de movimiento de las PU contiguas espaciales en el siguiente orden: A_1 , B_1 , B_0 , A_0 o B_2 .

[0142] A continuación, el codificador de vídeo 20 y/o el decodificador de vídeo 30 puede realizar una depuración restringida como sigue. Si el candidato de vector de movimiento en la ubicación A_1 y el candidato de MVI tienen los mismos vectores de movimiento y los mismos índices de referencia, el candidato de vector de movimiento en A_1 no se inserta en la lista de candidatos de fusión. Si el candidato de vector de movimiento en la ubicación B_1 y el candidato de vector de movimiento en la ubicación A_1 /candidato de MVI tienen los mismos vectores de movimiento y los mismos índices de referencia, el candidato de vector de movimiento en la ubicación B_1 no se inserta en la lista de candidatos de fusión. Si el candidato de vector de movimiento en la ubicación B_0 está disponible, el candidato de vector de movimiento en la ubicación B_0 se añade a la lista de candidatos de fusión. Si el candidato de vector de movimiento en la ubicación A_0 está disponible, el candidato de vector de movimiento en la ubicación A_0 se añade a la lista de candidatos de fusión. Si el candidato de vector de movimiento en la ubicación B_2 está disponible, el candidato de vector de movimiento en la ubicación B_2 se añade a la lista de candidatos de fusión.

[0143] El procedimiento de derivación para el candidato de fusión temporal en la codificación de profundidad de 3D-HEVC es similar al procedimiento de derivación de candidatos de fusión temporales en HEVC, donde se utiliza la información de movimiento de la PU coubicada. Sin embargo, en la codificación de profundidad de 3D-HEVC, se puede cambiar el índice de imagen de referencia objetivo del candidato de fusión temporal, como se explica anteriormente, en lugar de fijarse para que sea 0.

[0144] Ahora se describirá el procedimiento de derivación para candidatos de fusión bipredictivos combinados en la codificación de profundidad de 3D-HEVC. Si el número total de candidatos derivados de las dos etapas anteriores es menor que el número máximo de candidatos, se realiza el mismo procedimiento que se definió en HEVC, excepto para la especificación de $1OCandIIdx$ e $1ICandIIdx$. La relación entre $combIIdx$, $1OCandIIdx$ e $1ICandIIdx$ se define en la tabla ilustrada en la FIG. 12.

[0145] El procedimiento de derivación para candidatos de fusión de vector de movimiento cero en la codificación de profundidad de 3D-HEVC es el mismo que el procedimiento definido en HEVC.

[0146] Lo siguiente describe técnicas de ejemplo para la predicción residual avanzada (ARP). La ARP aplicada a las CU con modo de particionamiento igual a $Part_2Nx2N$ (por ejemplo, NxN en la FIG. 5) se adoptó en la 4.ª conferencia de JCT3V, como se propone en JCT3V-D0177. El documento JCT3V-D0177 se titula "CE4: Advanced residual prediction for multiview coding", por Zhang *et al.* El documento JCT3V-D0177 está disponible, a partir del 22 de agosto de 2014, en http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=862.

[0147] La FIG. 14 ilustra la estructura de predicción de la predicción residual avanzada (ARP) en la codificación de vídeo multivista. Como se muestra en la FIG. 14, los siguientes bloques se invocan en la predicción del bloque actual ("Curr") 140. El bloque de referencia 142 en la vista de referencia/básica 144 derivada por el vector de disparidad (DV) 146 se marca como "Base". El bloque 148 en la misma vista (vista V_m) que el bloque actual Curr 140, derivado por el vector de movimiento (temporal) 150 (indicado como TMV) del bloque actual 140 se marca como "CurrTRef". El bloque 152 en la misma vista que el bloque Base 142 (vista V_0), derivado por el vector de movimiento temporal (TMV) del bloque actual se marca como "BaseTRef". El bloque de referencia BaseTRef 152 se identifica con un vector de $TMV+DV$ 154 en comparación con el bloque actual Curr 140.

[0148] El predictor residual se indica como BaseTRef-Base, en el que la operación de resta se aplica a cada píxel de las matrices de píxeles indicadas. Un factor de ponderación "w" se puede multiplicar adicionalmente al predictor residual. Por lo tanto, el predictor final del bloque actual Curr se puede indicar como $CurrTRef + w * (BaseTRef - Base)$.

[0149] Cabe destacar que en las descripciones anteriores y la FIG. 14, se supone que se aplica la predicción unidireccional. Cuando se extiende a ARP a predicción bidireccional, se aplican las etapas anteriores para cada lista de imágenes de referencia. Cuando el bloque actual Curr usa una imagen de referencia entre vistas (en una vista diferente) para una de las dos listas de imágenes de referencia, se deshabilita el procedimiento de ARP.

[0150] Lo siguiente describe el procedimiento de descodificación en ARP. Primero, el descodificador de vídeo 30 obtiene un vector de disparidad (por ejemplo, usando el procedimiento de derivación de NBDV) que apunta a una vista de referencia objetivo. A continuación, en la imagen de la vista de referencia dentro de la misma unidad de acceso, el descodificador de vídeo 30 ubica el bloque correspondiente usando el vector de disparidad.

[0151] El descodificador de vídeo 30 puede reutilizar la información de movimiento del bloque actual para derivar la información de movimiento para el bloque de referencia. A continuación, el descodificador de vídeo 30 puede aplicar compensación de movimiento para el bloque correspondiente en base al mismo vector de movimiento del bloque actual, y la imagen de referencia derivada, para derivar un bloque residual.

[0152] La FIG. 15 es un diagrama conceptual que ilustra una relación de ejemplo entre el bloque actual 160, el bloque de referencia 162 y los bloques compensados por movimiento 164 y 166. La imagen de referencia en la vista de referencia (V_0) que tiene el mismo valor POC (recuento de orden de imágenes) que la imagen de referencia de la vista actual (V_m) se selecciona como la imagen de referencia del bloque 162 correspondiente. El codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden aplicar un factor de ponderación al bloque de residuos para obtener un bloque de residuos ponderado y añadir los valores del bloque de residuos ponderado a las muestras predichas.

[0153] Lo siguiente describe el factor de ponderación. Se usan tres factores de ponderación en ARP, es decir, 0, 0,5 y 1. El factor de ponderación que da lugar al coste de velocidad-distorsión mínimo para la CU actual se selecciona como el factor de ponderación final y el índice del factor de ponderación correspondiente (por ejemplo, 0, 1 y 2 que corresponden al factor de ponderación 0, 1 y 0,5, respectivamente) se transmite en el flujo de bits en el nivel de CU. En un ejemplo de ARP, todas las predicciones de PU en una CU comparten el mismo factor de ponderación. Cuando el factor de ponderación es igual a 0, la ARP no se usa para la CU actual.

[0154] Lo siguiente describe algunas simplificaciones adicionales para ARP. Primero, se describe la selección de imagen de referencia por medio de escalamiento del vector de movimiento. Segundo, se describe el filtro de interpolación.

[0155] Para la selección de imagen de referencia por medio del escalamiento del vector de movimiento, en 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. El documento JCT3V-C0049 se titula "3D-CE4: Advanced residual prediction for multiview coding [3D-CE4: predicción residual avanzada para codificación multivista]", por Zhang *et al.* El

documento JCT3V-C0049 está disponible, el 23 de septiembre de 2013, en http://phenix.int-evry.fr/jct3v/doc_end_user/current_document.php?id=487.

[0156] Por lo tanto, puede ser necesario acceder a diferentes imágenes de la vista de referencia para generar el bloque compensado por movimiento (por ejemplo, BaseTRef en la FIG. 14) del bloque correspondiente. Se ha propuesto ajustar a escala los vectores de movimiento descodificados de la PU actual hacia una imagen fija antes de realizar la compensación de movimiento para el procedimiento de generación residual cuando el factor de ponderación es diferente de 0. Como se propone en JCT3V-D0177, la imagen fija se define como la primera imagen de referencia de cada lista de imágenes de referencia si esta es desde la misma vista. Cuando el vector de movimiento descodificado no apunta a la imagen fija, el descodificador de vídeo 30 puede ajustar a escala primero el vector de movimiento descodificado y a continuación usar el vector de movimiento ajustado a escala para identificar CurrTRef y BaseTRef. Una imagen de referencia de este tipo usada para ARP se llama imagen de referencia de ARP objetivo.

[0157] Para los filtros de interpolación, como se describe en JCT3V-C0049, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden aplicar un filtro bilineal durante el procedimiento de interpolación del bloque correspondiente y su bloque de predicción. Para el bloque de predicción de la PU actual en las vistas no básicas, se puede aplicar un filtro convencional de 8/4 tomas. En otro ejemplo, como se propone en JCT3V-D0177, el codificador de vídeo 20 y/o el descodificador de vídeo 30 siempre pueden emplear filtrado bilineal independientemente de si el bloque está en vista básica o no básica cuando se aplica ARP.

[0158] En uno o más ejemplos de la divulgación, el codificador de vídeo 20 y/o el descodificador de vídeo 30 se pueden configurar para identificar la vista de referencia usando el índice de orden de vista devuelto por el procedimiento de derivación de NBDV. En un ejemplo de ARP, cuando la imagen de referencia de una PU en una lista de imágenes de referencia es de una vista diferente de la vista actual, ARP se deshabilita para esta lista de imágenes de referencia.

[0159] Algunas técnicas de adición para la codificación inter de profundidad se describen en las solicitudes provisionales de EE. UU. n.ºs 61/840.400, presentada el 27 de junio de 2013, y 61/847.942, presentada el 18 de julio de 2013. En estos ejemplos, cuando se codifica una imagen de profundidad, un vector de disparidad se convierte en un valor de profundidad estimado de las muestras contiguas del bloque actual.

[0160] En otros ejemplos para ARP, se pueden derivar candidatos de fusión adicionales, por ejemplo, accediendo al bloque de referencia de la vista básica identificada por un vector de disparidad.

[0161] Lo siguiente describe técnicas para ubicar un bloque para la predicción de movimiento entre vistas. En 3D-HEVC, un bloque 4x4 de referencia se identifica usando dos etapas generales. La primera etapa es identificar un píxel en una vista de referencia con un vector de movimiento de disparidad. La segunda etapa es obtener el bloque 4x4 correspondiente (con un conjunto único de información de movimiento correspondiente a RefPicList0 o RefPicList1, respectivamente) y utilizar la información de movimiento para crear un candidato de fusión.

[0162] El píxel (xRef, Yref) en la vista de referencia se identifica como sigue:

$$xRef = \text{Clip3}(0, \text{PicWidthInSamples}_L - 1, xP + ((nPSW - 1) \gg 1) + ((mvDisp[0] + 2) \gg 2)) \quad (\text{H-124})$$

$$yRef = \text{Clip3}(0, \text{PicHeightInSamples}_L - 1, yP + ((nPSH - 1) \gg 1) + ((mvDisp[1] + 2) \gg 2)) \quad (\text{H-125})$$

en el que (xP, yP) es la coordinación de la muestra superior izquierda de la PU actual, mvDisp es el vector de disparidad y nPSWxnPSH es el tamaño de la PU actual y PicWidthInSamples_L y PicHeightInSamples_L definen la resolución de la imagen en la vista de referencia (igual que la vista actual).

[0163] Lo siguiente describe la predicción de movimiento entre vistas a nivel sub-PU. En JCT3V-E0184, se propuso usar un procedimiento de predicción de movimiento entre vistas a nivel sub-PU para el IPMVC, es decir, el candidato derivado de un bloque de referencia en la vista de referencia. El JCT3V-E0184, "3D-CE3.h related: Sub-PU level inter-view motion prediction", de An et al., está disponible en http://phenix.it-sudparis.eu/jct2/docenduser/current_document.php?id=1198.

[0164] El concepto básico de predicción de movimiento entre vistas se describe anteriormente (por ejemplo, con respecto al procedimiento de derivación de candidatos entre vistas para el modo de salto/fusión), en el que solo se usa la información de movimiento del bloque de referencia asociado con la posición central para la PU actual en la vista dependiente. Sin embargo, la PU actual puede corresponder a un área de referencia (con el mismo tamaño que la PU actual identificada por el vector de disparidad) en la vista de referencia, y el área de referencia puede tener abundante información de movimiento (es decir, más que en el vector de movimiento).

[0165] Por lo tanto, se propone un procedimiento de predicción de movimiento entre vistas a nivel de sub-PU (SPIVMP). La FIG. 16 es un diagrama conceptual que ilustra la predicción de movimiento entre vistas de la sub-unidad de predicción (PU). Como se muestra en la FIG. 16, la PU actual 170 en la vista actual V1 se puede particionar en múltiples sub-PU (por ejemplo, cuatro sub-PU). Se puede usar un vector de disparidad para cada sub-PU para ubicar los bloques de referencia correspondientes en una vista de referencia V0. El codificador de vídeo 20 y/o el descodificador de vídeo 30 se pueden configurar para copiar (es decir, reutilizar) los vectores de movimiento asociados con cada uno de los bloques de referencia para su uso con las sub-PU correspondientes de la PU actual 170.

[0166] En un ejemplo, un candidato de fusión entre vistas temporal se deriva como sigue. Primero, se indica el tamaño de sub-PU asignado por $N \times N$. Primero, se divide la PU actual en múltiples sub-PU con un tamaño más pequeño. Se indica el tamaño de PU actual por $nPSW \times nPSH$ y el tamaño de sub-PU por $nPSW_{sub} \times nPSH_{sub}$.

$$nPSW_{sub} = \min(N, nPSW)$$

$$nPSH_{sub} = \min(N, nPSH)$$

[0167] Segundo, se establece un vector de movimiento por defecto $tmvLX$ en $(0, 0)$ y el índice de referencia $refLX$ en -1 para cada lista de imágenes de referencia. Para cada sub-PU en el orden de exploración de barrido, se aplica lo siguiente. Se añade el DV a la posición media de la sub-PU actual para obtener una ubicación de muestra de referencia ($xRefSub$, $yRefSub$) por:

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

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

[0168] Un bloque en la vista de referencia que abarca $(xRefSub, yRefSub)$ se usa como el bloque de referencia para la sub-PU actual.

[0169] Para el bloque de referencia identificado, si se codifica usando vectores de movimiento temporales, se aplica lo siguiente. Si tanto $refL0$ como $refL1$ son iguales a -1 y la sub-PU actual no es la primera en el orden de exploración de barrido, la información de movimiento del bloque de referencia se hereda por todas las sub-PU previas. Los parámetros de movimiento asociados se pueden usar como parámetros de movimiento candidatos para la sub-PU actual. Los elementos sintácticos $tmvLX$ y $refLX$ se actualizan a la información de movimiento de la sub-PU actual. De otro modo (por ejemplo, si el bloque de referencia está intracodificado), la información de movimiento de la sub-PU actual se establece en $tmvLX$ y $refLX$. Se pueden aplicar diferentes tamaños de bloques de sub-PU, por ejemplo, 4×4 , 8×8 y 16×16 . El tamaño de la sub-PU se puede señalar en VPS.

[0170] Lo siguiente describe la herencia de vector de movimiento a nivel sub-PU para la codificación de profundidad. De forma similar a las propuestas para la predicción de movimiento entre vistas a nivel sub-PU de una vista de textura a otra vista de textura, la provisional de EE. UU. 61/858.089, presentada el 24 de julio de 2013, propuso técnicas que aplican la predicción de movimiento a nivel sub-PU de una vista de textura a la vista de profundidad correspondiente. Es decir, una PU actual se puede particionar en varias sub-PU y cada sub-PU usa la información de movimiento de un bloque de textura coubicado para la compensación de movimiento. En este caso, se admite MVI a nivel de sub-PU y el vector de disparidad usado por la predicción de movimiento entre vistas se considera que es siempre cero.

[0171] El diseño actual para BVSP en 3D-HEVC presenta los siguientes problemas. Cuando se usa AMP y el tamaño de PU actual es 4×16 o 16×4 , y la PU se predice unidireccionalmente, BVSP se logra derivando un vector de disparidad para la PU completa. Es decir, cada subbloque en la PU usa el mismo vector de disparidad para la síntesis de bloque de referencia y la compensación de movimiento. Como tal, para tamaños de bloque más grandes, BVSP puede ser menos eficaz porque usa el mismo vector de disparidad para la síntesis de bloque y la compensación de movimiento para todos los subbloques puede ser menos óptima para algunos de los subbloques.

[0172] Como otro inconveniente, cuando la PU actual se predice bidireccionalmente, BVSP se habilita con tamaños de bloque iguales a 4×8 y 8×4 . Sin embargo, en HEVC, la compensación de movimiento para bloques que contienen menos de 64 píxeles (por ejemplo, bloques de 4×8 u 8×4) no está permitida (aunque se permiten compensaciones de movimiento de 16×4 y 4×16).

[0173] En vista de estos inconvenientes, la presente divulgación propone técnicas relacionadas con la predicción de síntesis de vista en 3D-HEVC, centrándose en tamaños de compensación de movimiento de BVSP. De acuerdo con las técnicas de la presente divulgación, para BVSP, cada PU se puede particionar en subbloques, y cada subbloque se puede asociar con un vector de movimiento de disparidad diferente y compensarse por movimiento por separado. De esta manera, se puede incrementar la exactitud de BVSP para bloques separados con AMP, y por tanto, se puede

incrementar la eficacia de codificación. De acuerdo con las técnicas de la divulgación, el tamaño de los subbloques disponibles para su uso con BVSP se puede definir además como sigue.

5 **[0174]** En un ejemplo de la divulgación, cuando el tamaño de PU actual es 16x4 (o 4x16), y la PU actual se predice unidireccionalmente, el codificador de vídeo 20 y/o el descodificador de vídeo 30 se pueden configurar para aplicar técnicas de BVSP y compensación de movimiento para subbloques de 8x4 (o 4x8) de la PU actual. Es decir, el tamaño de la subregión de BVSP es 8x4 (o 4x8). Cada uno de los subbloques se puede asignar con un vector de movimiento de disparidad convertido de un bloque de profundidad.

10 **[0175]** La FIG. 17 es un dibujo conceptual que representa técnicas de BVSP y compensación de movimiento de la presente divulgación cuando se usa AMP. En el ejemplo de la FIG. 17, el codificador de vídeo 20 y/o el descodificador de vídeo 30 separa asimétricamente la PU actual 250 en un bloque 4x16. Cabe destacar que la particionamiento 4x16 es solo un ejemplo, y que las técnicas de la presente divulgación descritas con referencia a la FIG. 17 se pueden aplicar a otras separaciones asimétricas, incluyendo las separaciones 16x4. El codificador de vídeo 20 y/o el descodificador de vídeo 30 se pueden configurar para subdividir la PU 250 en subbloques 4x8 255 y 256.

15 **[0176]** En el ejemplo de la FIG. 17, el codificador de vídeo 20 y/o el descodificador de vídeo 30 están configurados para predecir unidireccionalmente la PU 250 usando BVSP. A este respecto, el codificador de vídeo 20 y/o el descodificador de vídeo 30 se pueden configurar para derivar un vector de disparidad para PU 250, por ejemplo, usando técnicas de derivación de NBDV. Por ejemplo, el vector de disparidad 261 se puede derivar del bloque contiguo 252. A continuación, el codificador de vídeo 20 y/o el descodificador de vídeo 30 se pueden configurar para reutilizar el vector de disparidad 261 para ubicar un bloque de profundidad 260 correspondiente en una imagen de profundidad de referencia. De acuerdo con las técnicas de la presente divulgación, en lugar de usar la totalidad del bloque de profundidad 260 para derivar un único vector de movimiento de disparidad para PU 255, el codificador de vídeo 20 y/o el descodificador de vídeo 30 se pueden configurar para derivar un vector de movimiento de disparidad de subbloque 4x8 264 del bloque de profundidad 260 para el subbloque 255, y para derivar un vector de movimiento de disparidad de subbloque 4x8 262 del bloque de profundidad 26 para el subbloque 256.

20 **[0177]** A continuación, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden sintetizar un bloque de referencia para cada uno de los subbloques 255 y 256 usando los vectores de movimiento de disparidad derivados correspondientes para realizar la compensación de movimiento con la imagen de referencia correspondiente a la imagen de referencia entre vistas con un orden de vista igual a refVldxLX. Al derivar vectores de movimiento de disparidad individuales para cada uno de los subbloques 255 y 256, se pueden sintetizar vistas de referencia más exactas y se puede lograr el procedimiento de compensación de movimiento correspondiente con ganancias de codificación incrementadas.

25 **[0178]** En otro ejemplo de la divulgación, cuando el tamaño de la PU actual es 16x12 (o 12x16), y la PU actual se predice unidireccionalmente, el codificador de vídeo 20 y/o el descodificador de vídeo 30 se pueden configurar para particionar la PU actual en subbloques 8x4 (o 4x8) (también llamados subregiones de BVSP) y derivar un vector de movimiento de disparidad para cada subbloque usando BVSP.

30 **[0179]** En otro ejemplo, el tamaño de las subregiones de BVS se puede asignar a 16x12 o 12x16. Todavía en otro ejemplo, cada subbloque 16x12 (o 12x16) se separa además en un subbloque 16x8 (u 8x16) y dos subbloques 8x4 (o 4x8) que son adyacentes a la PU 16x4 (4x16) en la misma CU. En otro ejemplo, los subbloques 16x8 (u 8x16) se pueden particionar además en dos subregiones 8x8 o cuatro subregiones 4x8 (u 8x4) en base, por ejemplo, a las 4 esquinas del bloque de profundidad correspondiente.

35 **[0180]** En otro ejemplo de la divulgación, cuando tanto la altura como el ancho de la PU actual son mayores que o iguales a 8, y la PU se predice bidireccionalmente, el codificador de vídeo 20 y/o el descodificador de vídeo 30 están configurados para establecer el tamaño de la subregión de BVSP en 8x8 en lugar de 4x8 u 8x4, como en propuestas previas para 3D-HEVC. En otro ejemplo, en lugar de usar BVSP bipredictiva para las PU con tamaños iguales a 12x16 o 16x12, se puede aplicar BVSP unipredictiva. En este caso, el tamaño de compensación de movimiento se puede establecer además en 4x16 o 16x4. En otro ejemplo, cuando el tamaño de la PU actual es 16x4 o 4x16, y la PU actual se predice bidireccionalmente, el tamaño de la subregión de BVSP se establece igual al tamaño de la PU.

40 **[0181]** La predicción de movimiento de sub-PU puede presentar los siguientes inconvenientes. En este contexto, la predicción de movimiento de sub-PU puede incluir las técnicas de predicción de movimiento de sub-PU propuestas en JCT3V-E0184, como se describe anteriormente, así como la extensión de predicción de movimiento de sub-PU a MVI desde la vista de textura a la vista de profundidad.

45 **[0182]** Como un inconveniente, cuando se habilita la particionamiento de movimiento asimétrica (AMP), el tamaño de PU actual es igual, por ejemplo, a 4x16, 16x4, y el tamaño de bloque de sub-PU señalado en el VPS es igual a 8x8, en base a propuestas previas para la predicción de movimiento de sub-PU, dichas PU se particionarán en dos subbloques 4x8 u 8x4. Para cada subbloque, se hereda la información de movimiento de un bloque de referencia. La información de movimiento se puede identificar por un vector de disparidad en una vista de textura de referencia para la predicción de movimiento entre vistas o se puede reutilizar de un bloque de textura coubicado en la vista de textura

correspondiente para la herencia del vector de movimiento. En este ejemplo, se puede invocar bipredicción basada en 4x8 u 8x4, lo que no está permitido por HEVC.

5 **[0183]** Como otro inconveniente, cuando se habilita AMP, el tamaño de PU es igual a, por ejemplo, 12x16 o 16x12, y el tamaño de bloque de sub-PU señalado (es decir, tamaño de subbloque) en VPS es igual a 8x8, en base a propuestas previas para la predicción de movimiento de sub-PU, dichas PU se particionarán en dos subbloques 8x8 y dos 4x8/8x4. De forma similar al caso anterior, los subbloques 4x8/8x4 pueden usar bipredicción, lo que no está permitido por HEVC.

10 **[0184]** En la presente divulgación se proponen técnicas relacionadas con la predicción de movimiento entre vistas y la herencia del vector de movimiento (para una PU de profundidad). Las técnicas de la presente divulgación se pueden aplicar en el contexto cuando un índice de fusión indica predicción de movimiento entre vistas o MVI. En particular, las técnicas de predicción de movimiento entre vistas y/o MVI de la divulgación incluyen técnicas para la particionamiento adicional de las PU de AMP en subbloques y la obtención de información de movimiento separada para cada uno de los subbloques. De esta manera, la exactitud de la predicción de movimiento entre vistas y/o MVI se puede mejorar para cada uno de los subbloques, y por tanto, se puede incrementar la eficacia de codificación.

15 **[0185]** En un ejemplo de la divulgación, cuando una PU actual se codifica usando predicción de movimiento entre vistas y/o MVI, y el tamaño de PU actual es igual a 4x16 o 16x4, el codificador de vídeo 20 y/o el descodificador de vídeo 30 se pueden configurar para particionar la PU en dos subbloques de 4x8 u 8x4. Para cada uno de los subbloques, el codificador de vídeo 20 y/o el descodificador de vídeo 30 se pueden configurar para obtener solo información de movimiento de un bloque de referencia correspondiente a una lista de imágenes de referencia particular (por ejemplo, RefPicList0). La información de movimiento correspondiente a un bloque de referencia en RefPicList0 se hereda para los subbloques 4x8 u 8x4. En este caso, los subbloques se predicen unidireccionalmente de una imagen en RefPicList0.

20 **[0186]** La FIG. 18 es un diagrama conceptual que ilustra la herencia del vector de movimiento y las técnicas de compensación de movimiento para las PU separadas asimétricamente en tamaños 4x16 y 16x4. Por ejemplo, para una PU 4x16, el codificador de vídeo 20 y/o el descodificador de vídeo 30 se codifican para dividir adicionalmente la PU 4x16 en dos subbloques 4x8 300 y 302. La información de movimiento para cada uno de los subbloques 300 y 302 se obtiene de un bloque de referencia en una imagen de referencia que pertenece a una lista de imágenes de referencia particular (por ejemplo, RefPicList0). A continuación, se realiza la compensación de movimiento para cada uno de los subbloques 300 y 302 con respecto a un bloque de referencia en RefPicList0. Del mismo modo, para una PU 16x4, el codificador de vídeo 20 y/o el descodificador de vídeo 30 se configuran para dividir adicionalmente la PU 16x4 en dos subbloques 8x4 304 y 306. La información de movimiento para cada uno de los subbloques 304 y 306 se obtiene de un bloque de referencia en una imagen de referencia que pertenece a una lista de imágenes de referencia particular (por ejemplo, RefPicList0). A continuación, se realiza la compensación de movimiento para cada uno de los subbloques 304 y 306 con respecto a un bloque de referencia en RefPicList0.

30 **[0187]** En otro ejemplo de la divulgación, cuando el tamaño de PU actual es uno de 16x12, 12x16, 4x16 o 16x4, el codificador de vídeo 20 y el descodificador de vídeo 30 se configuran para no usar bipredicción para subbloques 8x4/4x8 cuando se aplica la predicción de movimiento entre vistas y/o MVI (para profundidad) a nivel de sub-PU. Es decir, cuando el tamaño de PU actual es uno de 16x12, 12x16, 4x16 o 16x4, el codificador de vídeo 20 y el descodificador de vídeo 30 se configuran para usar solo unipredicción para subbloques 8x4/4x8 cuando se aplica la predicción de movimiento entre vistas y/o MVI (para profundidad) a nivel de sub-PU.

35 **[0188]** En otro ejemplo de la divulgación, se propone que cuando se aplica la predicción de movimiento entre vistas o MVI a nivel de sub-PU, y el tamaño de PU actual es igual a 4x16 o 16x4, la PU no se separa en sub-PU.

40 **[0189]** En otro ejemplo de la divulgación, se propone que cuando se aplica la predicción de movimiento entre vistas o MVI a nivel de sub-PU, y el tamaño de PU actual es igual a 12x16 o 16x12, la PU se separa en tres bloques de sub-PU de igual tamaño con un tamaño igual a 4x16 o 16x4. Para cada bloque de sub-PU, se hereda la información de movimiento del bloque de referencia correspondiente.

45 **[0190]** En otro ejemplo de la divulgación, cuando el tamaño de PU actual es igual a 12x16 o 16x12, la PU se separa en dos bloques de sub-PU 8x8 y uno 4x16 o 16x4, en los que las sub-PU 8x8 forman la mitad izquierda o superior de la CU que contiene esta PU. En otro aspecto de este ejemplo, los subbloques 4x16 y 16x4 se separan adicionalmente en dos bloques de sub-PU 4x8 u 8x4. Para cada sub-PU 4x8 u 8x4, solo se obtiene la información de movimiento de un bloque de referencia correspondiente a la lista de imágenes de referencia (RefPicList0) y se reutiliza para la sub-PU 4x8 u 8x4. En este caso, las sub-PU se predicen unidireccionalmente de una imagen en RefPicList0.

50 **[0191]** En otro ejemplo de la divulgación, cuando se usa BVSP para una PU con un tamaño igual a 12x16 o 16x12, la PU se separa en tres bloques de sub-PU de igual tamaño con tamaños iguales a 4x16 o 16x4. A continuación, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden derivar un vector de movimiento de disparidad único para cada sub-PU del bloque de profundidad correspondiente.

55 **[0191]** En otro ejemplo de la divulgación, cuando se usa BVSP para una PU con un tamaño igual a 12x16 o 16x12, la PU se separa en tres bloques de sub-PU de igual tamaño con tamaños iguales a 4x16 o 16x4. A continuación, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden derivar un vector de movimiento de disparidad único para cada sub-PU del bloque de profundidad correspondiente.

60 **[0191]** En otro ejemplo de la divulgación, cuando se usa BVSP para una PU con un tamaño igual a 12x16 o 16x12, la PU se separa en tres bloques de sub-PU de igual tamaño con tamaños iguales a 4x16 o 16x4. A continuación, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden derivar un vector de movimiento de disparidad único para cada sub-PU del bloque de profundidad correspondiente.

65 **[0191]** En otro ejemplo de la divulgación, cuando se usa BVSP para una PU con un tamaño igual a 12x16 o 16x12, la PU se separa en tres bloques de sub-PU de igual tamaño con tamaños iguales a 4x16 o 16x4. A continuación, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden derivar un vector de movimiento de disparidad único para cada sub-PU del bloque de profundidad correspondiente.

[0192] La FIG. 19 es un diagrama de bloques que ilustra un codificador de vídeo 20 de ejemplo que puede implementar las técnicas de la presente divulgación. El codificador de vídeo 20 puede realizar intra e intercodificación (incluyendo la codificación entre vistas) de bloques de vídeo dentro de fragmentos de vídeo, por ejemplo, fragmentos tanto de imágenes de textura como de mapas de profundidad. La información de textura, en general, incluye información de luminancia (brillo o intensidad) y crominancia (color, por ejemplo, tonos rojos y tonos azules). En general, el codificador de vídeo 20 puede determinar los modos de codificación con respecto a los fragmentos de luminancia, y reutilizar la información de predicción de la codificación de la información de luminancia para codificar la información de crominancia (por ejemplo, reutilizando la información de particionamiento, las selecciones de modo de predicción intra, los vectores de movimiento o similares). La intracodificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial de un vídeo dentro de una trama o imagen de vídeo dada. La intercodificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal de un vídeo dentro de tramas o imágenes adyacentes de una secuencia de vídeo. El modo intra (modo I) se puede referir a cualquiera de varios modos de codificación de base espacial. Los modos inter, tales como la predicción unidireccional (modo P) o la bipredicción (modo B), se pueden referir a cualquiera de varios modos de codificación de base temporal.

[0193] Como se muestra en la FIG. 19, el codificador de vídeo 20 recibe un bloque de vídeo actual (es decir, un bloque de datos de vídeo, tal como un bloque de luminancia, un bloque de crominancia o un bloque de profundidad) dentro de una trama de vídeo (por ejemplo, una imagen de textura o un mapa de profundidad) que se va a codificar. En el ejemplo de la FIG. 19, el codificador de vídeo 20 incluye la memoria de datos de vídeo 40, la unidad de selección de modo 41, la memoria intermedia de imágenes descodificadas (DPB) 64, el sumador 50, la unidad de procesamiento de transformada 52, la unidad de cuantificación 54, la unidad de filtro de bucle 63 y la unidad de codificación de entropía 56. A su vez, la unidad de selección de modo 41 incluye la unidad de compensación de movimiento 44, la unidad de estimación de movimiento 42, la unidad de procesamiento de predicción intra 46 y la unidad de particionamiento 48. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 también incluye la unidad de cuantificación inversa 58, la unidad de procesamiento de transformada inversa 60 y el sumador 62. La unidad de filtro de bucle 63 puede incluir un filtro de eliminación de bloques y un filtro SAO para filtrar las fronteras del bloque para eliminar distorsiones de pixelado del vídeo reconstruido. También se pueden usar filtros adicionales (en el bucle o tras el bucle), además del filtro de eliminación de bloques. Dichos filtros no se muestran por razones de brevedad, pero si se desea, pueden filtrar la salida del sumador 50 (como un filtro en el bucle).

[0194] La memoria de datos de vídeo 40 puede almacenar datos de vídeo que se van 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, del origen de vídeo 18. La DPB 64 es una memoria intermedia que almacena datos de vídeo de referencia para su uso en la codificación de datos de vídeo por el codificador de vídeo 20 (por ejemplo, en modos de intra o intercodificación, también denominados modos de codificación de predicción intra o inter). La memoria de datos de vídeo 40 y la DPB 64 pueden estar formadas por cualquiera de una variedad de 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 DPB 64 se pueden proporcionar por el mismo dispositivo de memoria o por dispositivos de memoria separados. 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 del chip con respecto a esos componentes.

[0195] Durante el procedimiento de codificación, el codificador de vídeo 20 recibe una trama o un fragmento de vídeo que se va a codificar. La trama o el fragmento se pueden dividir en múltiples bloques de vídeo. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 realizan la codificación interpredictiva del bloque de vídeo recibido con respecto a uno o más bloques en una o más tramas de referencia, incluyendo las tramas de referencia entre vistas, para proporcionar predicción temporal. La unidad de procesamiento de predicción intra 46, de forma alternativa, puede realizar la codificación intrapredictiva del bloque de vídeo recibido con respecto a uno o más bloques contiguos en la misma trama o fragmento que el bloque que se va a codificar para proporcionar predicción espacial. El codificador de vídeo 20 puede realizar múltiples pases de codificación, por ejemplo, para seleccionar un modo de codificación apropiado para cada bloque de datos de vídeo.

[0196] Además, la unidad de particionamiento 48 puede particionar bloques de datos de vídeo en subbloques, en base a la evaluación de esquemas de particionamiento previos en pases de codificación previos. Por ejemplo, la unidad de particionamiento 48 puede particionar inicialmente una trama o un fragmento en LCU, y particionar cada una de las LCU en sub-CU en base a un análisis de velocidad-distorsión (por ejemplo, una optimización de velocidad-distorsión). La unidad de selección de modo 41 puede producir además una estructura de datos de árbol cuaternario, indicativa de la particionamiento de una LCU en las sub-CU. Las CU de nodos hojas del árbol cuaternario pueden incluir una o más PU y una o más TU.

[0197] La unidad de selección de modo 41 puede seleccionar uno de los modos de codificación, intra o inter, por ejemplo, en base a resultados de error, y proporciona el bloque intra o intercodificado resultante al sumador 50 para generar datos de bloque residuales y al sumador 62 para reconstruir el bloque codificado para su uso como una trama de referencia. La unidad de selección de modo 41 también proporciona elementos sintácticos, tales como vectores de movimiento, indicadores de modo intra, información de particionamiento y otra información sintáctica de este tipo, a la unidad de codificación por entropía 56.

- 5 **[0198]** La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden estar altamente integradas, pero se ilustran por separado para propósitos conceptuales. La estimación del movimiento, realizada por la unidad de estimación de movimiento 42, es el procedimiento de generación de vectores de movimiento, que estiman el movimiento para los bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de una PU de un bloque de vídeo dentro de una trama o imagen de vídeo actual con respecto a un bloque predictivo dentro de una trama de referencia (u otra unidad codificada) con respecto al bloque actual que se está codificando dentro de la trama actual (u otra unidad codificada).
- 10 **[0199]** Un bloque predictivo es un bloque que se encuentra que coincide estrechamente con el bloque que se va a codificar, en términos de la diferencia de píxeles, lo que se puede determinar por la suma de la diferencia absoluta (SAD), la suma de la diferencia de cuadrados (SSD) u otras métricas de diferencia. En algunos ejemplos, el codificador de vídeo 20 puede calcular valores para posiciones de píxeles fraccionarios de las imágenes de referencia almacenadas en la DPB 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 de píxel fraccionario de la imagen de referencia. Por lo tanto, la unidad de estimación de movimiento 42 puede realizar una búsqueda de movimiento con respecto a las posiciones de píxel completo y las posiciones de píxel fraccionario y emitir un vector de movimiento con una precisión de píxel fraccionario.
- 15 **[0200]** La unidad de estimación de movimiento 42 calcula un vector de movimiento para una PU de un bloque de vídeo en un fragmento intercodificado, comparando la posición de la PU con la posición de un bloque predictivo de una imagen de referencia. La imagen de referencia se puede seleccionar de una primera lista de imágenes de referencia (lista 0) o una segunda lista de imágenes de referencia (lista 1), de las que cada una identifica una o más imágenes de referencia almacenadas en la DPB 64. Las listas de imágenes de referencia se pueden construir usando las técnicas de la presente divulgación. La unidad de estimación de movimiento 42 envía el vector de movimiento calculado a la unidad de codificación por entropía 56 y a la unidad de compensación de movimiento 44.
- 20 **[0201]** La compensación de movimiento, realizada por la unidad de compensación de movimiento 44, puede implicar capturar o generar el bloque predictivo en base al vector de movimiento determinado por la unidad de estimación de movimiento 42. De nuevo, la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 se pueden integrar funcionalmente, en algunos ejemplos. Tras recibir el vector de movimiento para la PU del bloque de vídeo actual, la unidad de compensación de movimiento 44 puede ubicar el bloque predictivo al que apunta el vector de movimiento en una de las listas de imágenes de referencia. El sumador 50 forma un bloque de vídeo residual restando los valores de píxel del bloque predictivo a los valores de píxel del bloque de vídeo actual que se está codificando, formando valores de diferencias de píxel, como se analiza a continuación. En general, la unidad de estimación de movimiento 42 realiza la estimación de movimiento con respecto a los componentes de luma, y la unidad de compensación de movimiento 44 usa los vectores de movimiento calculados en base a los componentes de luma, tanto para los componentes de croma como para los componentes de luma. De esta manera, la unidad de compensación de movimiento 44 puede reutilizar la información de movimiento determinada para los componentes de luma para codificar componentes de croma, de modo que la unidad de estimación de movimiento 42 no necesita realizar una búsqueda de movimiento para los componentes de croma. La unidad de selección de modo 41 también puede generar elementos sintácticos asociados con los bloques de vídeo y el fragmento de vídeo para su uso por el descodificador de vídeo 30 en la descodificación de los bloques de vídeo del fragmento de vídeo.
- 25 **[0202]** La unidad de procesamiento de predicción intra 46 puede intraprededir un bloque actual, de forma alternativa a la predicción inter realizada por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44, como se describe anteriormente. En particular, la unidad de procesamiento de predicción intra 46 puede determinar un modo de predicción intra para usar para codificar un bloque actual. En algunos ejemplos, la unidad de procesamiento de predicción intra 46 puede codificar un bloque actual usando diversos modos de predicción intra, por ejemplo, durante pases de codificación separados, y la unidad de procesamiento de predicción intra 46 (o la unidad de selección de modo 41, en algunos ejemplos) puede seleccionar un modo de predicción intra apropiado para usar de los modos sometidos a prueba.
- 30 **[0203]** Por ejemplo, la unidad de procesamiento de predicción intra 46 puede calcular valores de velocidad-distorsión usando un análisis de velocidad-distorsión para las diversos modos de predicción intra sometidos a prueba, y seleccionar el modo de predicción intra que tenga las mejores características de velocidad-distorsión entre los modos sometidos a prueba. El análisis de velocidad-distorsión determina, en general, una cantidad de distorsión (o error) entre un bloque codificado y un bloque original no codificado que se codificó para producir el bloque codificado, así como una velocidad de bits (es decir, un número de bits) usada para producir el bloque codificado. La unidad de procesamiento de predicción intra 46 puede calcular proporciones de las distorsiones y velocidades para los diversos bloques codificados, para determinar qué modo de predicción intra presenta el mejor valor de velocidad-distorsión para el bloque.
- 35 **[0204]** Después de seleccionar un modo de predicción intra para un bloque, la unidad de procesamiento de predicción intra 46 puede proporcionar información indicativa del modo de predicción intra seleccionado para el bloque a la unidad de codificación por entropía 56. La unidad de codificación por entropía 56 puede codificar la información que indica el
- 40
- 45
- 50
- 55
- 60
- 65

modo de predicción intra seleccionado. El codificador de vídeo 20 puede incluir, en el flujo de bits transmitido, datos de configuración que pueden incluir una pluralidad de tablas de índices del modo de predicción intra y una pluralidad de tablas de índices del modo de predicción intra modificadas (también denominadas tablas de correlación de palabras de código), definiciones de contextos de codificación para diversos bloques e indicaciones de un modo de predicción intra más probable, una tabla de índices del modo de predicción intra y una tabla de índices del modo de predicción intra modificada para usar para cada uno de los contextos.

[0205] El codificador de vídeo 20 forma un bloque de vídeo residual restando los datos de predicción de la unidad de selección de modo 41 del bloque de vídeo original que se está codificando. El sumador 50 representa el componente o componentes que realizan esta operación de resta. La unidad de procesamiento de transformada 52 aplica una transformada, tal como una transformada discreta de coseno (DCT) o una transformada conceptualmente similar, al bloque residual, produciendo un bloque de vídeo que comprende valores de coeficientes de transformada residuales. La unidad de procesamiento de transformada 52 puede realizar otras transformadas que son conceptualmente similares a la DCT. También se podrían usar transformadas de ondículas, transformadas de números enteros, transformadas de subbandas u otros tipos de transformadas. En cualquier caso, la unidad de procesamiento de transformada 52 aplica la transformada al bloque residual, produciendo un bloque de coeficientes de transformada residuales.

[0206] La transformada puede convertir la información residual desde un dominio de valores de píxel a un dominio de transformada, tal como un dominio de frecuencia. La unidad de procesamiento de transformada 52 puede enviar los coeficientes de transformada resultantes a la unidad de cuantificación 54. La unidad de cuantificación 54 cuantifica los coeficientes de transformada para reducir adicionalmente la velocidad de bits. El procedimiento de cuantificación puede reducir la profundidad de bits asociada con algunos de, o todos, los coeficientes. El grado de cuantificación se puede modificar ajustando un parámetro de cuantificación. En algunos ejemplos, la unidad de cuantificación 54 puede realizar a continuación una exploración de la matriz que incluye los coeficientes de transformada cuantificados. De forma alternativa, la unidad de codificación por entropía 56 puede realizar la exploración.

[0207] Tras la cuantificación, la unidad de codificación por entropía 56 codifica por entropía los coeficientes de transformada cuantificados. Por ejemplo, la unidad de codificación por entropía 56 puede realizar 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 sintaxis (SBAC), la codificación por entropía por particionamiento en intervalos de probabilidad (PIPE) u otra técnica de codificación por entropía. En el caso de la codificación por entropía basada en contexto, el contexto se puede basar en bloques contiguos. Tras la codificación por entropía por la unidad de codificación por entropía 56, el flujo de bits codificado se puede transmitir a otro dispositivo (por ejemplo, el descodificador de vídeo 30) o archivar para su posterior transmisión o recuperación.

[0208] La unidad de cuantificación inversa 58 y la unidad de procesamiento de transformada inversa 60 aplican la cuantificación inversa y la transformación inversa, respectivamente, para reconstruir el bloque residual en el dominio de píxel, por ejemplo, para su posterior uso como bloque de referencia. La unidad de compensación de movimiento 44 puede calcular un bloque de referencia añadiendo el bloque residual a un bloque predictivo de una de las tramas de la DPB 64. La unidad de compensación de movimiento 44 también puede aplicar uno o más filtros de interpolación al bloque residual reconstruido para calcular valores de píxeles fraccionarios, para su uso en la estimación del movimiento. El sumador 62 añade el bloque residual reconstruido al bloque de predicción de movimiento compensado por movimiento producido por la unidad de compensación de movimiento 44 para producir un bloque de vídeo reconstruido para su almacenamiento en la DPB 64. El bloque de vídeo reconstruido se puede usar por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44, como bloque de referencia para intercodificar un bloque en una trama de vídeo posterior.

[0209] El codificador de vídeo 20 puede codificar mapas de profundidad de una manera que se asemeja sustancialmente a las técnicas de codificación para codificar componentes de luminancia, si bien sin los correspondientes componentes de crominancia. Por ejemplo, la unidad de procesamiento de predicción intra 46 puede intraprededir bloques de mapas de profundidad, mientras que la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden interprededir bloques de mapas de profundidad. Sin embargo, como se analiza anteriormente, durante la predicción inter de mapas de profundidad, la unidad de compensación de movimiento 44 puede ajustar a escala (es decir, ajustar) los valores de los mapas de profundidad de referencia en base a las diferencias en los rangos de profundidad y los valores de precisión para los rangos de profundidad. Por ejemplo, si diferentes valores de profundidad máxima en el mapa de profundidad actual y un mapa de profundidad de referencia corresponden a la misma profundidad del mundo real, el codificador de vídeo 20 puede ajustar a escala el valor de profundidad máxima del mapa de profundidad de referencia para que sea igual al valor de profundidad máxima en el mapa de profundidad actual, para propósitos de predicción. Adicionalmente o de forma alternativa, el codificador de vídeo 20 puede usar los valores de rango de profundidad y los valores de precisión actualizados para generar una imagen de síntesis de vista para la predicción de síntesis de vista, por ejemplo, usando técnicas sustancialmente similares a la predicción entre vistas.

[0210] Como se analizará con más detalle a continuación con referencia a las FIGS. 21-23, el codificador de vídeo 20 se puede configurar para emplear las técnicas de la presente divulgación descritas anteriormente. En particular, el

codificador de vídeo 20 se puede configurar para particionar las PU en subbloques cuando dichas PU se separan de acuerdo con un modo de particionamiento asimétrica. A continuación, el codificador de vídeo 20 se puede configurar para heredar y/o derivar vectores de movimiento o vectores de movimiento de disparidad para cada uno de los subbloques.

5 [0211] La FIG. 20 es un diagrama de bloques que ilustra un ejemplo de descodificador de vídeo 30 que puede implementar las técnicas de la presente divulgación. En el ejemplo de la FIG. 20, el descodificador de vídeo 30 incluye la memoria de datos de vídeo 79, la unidad de descodificación por entropía 70, la unidad de compensación de movimiento 72, la unidad de procesamiento de predicción intra 74, la unidad de cuantificación inversa 76, la unidad de procesamiento de transformada inversa 78, la memoria intermedia de imágenes descodificadas (DPB) 82, la unidad de filtro de bucle 83 y el sumador 80. En algunos ejemplos, el descodificador de vídeo 30 puede realizar un pase de descodificación, en general, recíproco al pase de codificación descrito con respecto al codificador de vídeo 20 (FIG. 19). La unidad de compensación de movimiento 72 puede generar datos de predicción en base a vectores de movimiento recibidos desde la unidad de descodificación por entropía 70, mientras que la unidad de procesamiento de predicción intra 74 puede generar datos de predicción en base a indicadores de modo de predicción intra recibidos desde la unidad de descodificación por entropía 70.

20 [0212] La memoria de datos de vídeo 79 puede almacenar datos de vídeo, tales como un flujo de bits de vídeo codificado, que se va a descodificar por los componentes del descodificador de vídeo 30. Los datos de vídeo almacenados en la memoria de datos de vídeo 79 se pueden obtener desde un origen de vídeo local, tal como una cámara, por medio de comunicación de red por cable o inalámbrica de datos de vídeo, o al acceder a medios físicos de almacenamiento de datos. La memoria de datos de vídeo 79 puede formar una memoria intermedia de imágenes codificadas (CPB) que almacena datos de vídeo codificados a partir de un flujo de bits de vídeo codificado. La DPB 82 es un ejemplo de DPB que almacena datos de vídeo de referencia para su uso en la descodificación de datos de vídeo por el descodificador de vídeo 30 (por ejemplo, en los modos de intra o intercodificación, también denominados modos de codificación de predicción intra o inter). La memoria de datos de vídeo 79 y la DPB 82 pueden estar formadas por cualquiera de una variedad de 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 79 y la DPB 82 se pueden proporcionar por el mismo dispositivo de memoria o por dispositivos de memoria separados. En diversos ejemplos, la memoria de datos de vídeo 79 puede estar en un chip con otros componentes del descodificador de vídeo 30, o fuera del chip con respecto a esos componentes.

35 [0213] Durante el procedimiento de descodificación, el descodificador de vídeo 30 recibe un flujo de bits de vídeo codificado que representa los bloques de vídeo de un fragmento de vídeo codificado y elementos sintácticos asociados desde el codificador de vídeo 20. La unidad de descodificación por entropía 70 del descodificador de vídeo 30 descodifica por entropía el flujo de bits para generar coeficientes cuantificados, vectores de movimiento o indicadores de modo de predicción intra y otros elementos sintácticos. La unidad de descodificación por entropía 70 reenvía los vectores de movimiento y otros elementos sintácticos a la unidad de compensación de movimiento 72. El descodificador de vídeo 30 puede recibir los elementos sintácticos a nivel de fragmento de vídeo y/o a nivel de bloque de vídeo.

45 [0214] Cuando el fragmento de vídeo se codifica como un fragmento intracodificado (I), la unidad de procesamiento de predicción intra 74 puede generar datos de predicción para un bloque de vídeo del fragmento de vídeo actual, en base a un modo de predicción intra señalado y datos de bloques previamente descodificados de la trama o imagen actual. Cuando la trama de vídeo se codifica como un fragmento intercodificado (es decir, B, P o GPB), la unidad de compensación de movimiento 72 genera bloques predictivos para un bloque de vídeo del fragmento de vídeo actual basándose en los vectores de movimiento y en otros elementos sintácticos recibidos desde la unidad de descodificación 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 descodificador de vídeo 30 puede construir las listas de imágenes de referencia, la lista 0 y la lista 1, usando técnicas de la presente divulgación, en base a las imágenes de referencia almacenadas en la memoria intermedia de imágenes descodificadas 82. La unidad de compensación de movimiento 72 determina la información de predicción para un bloque de vídeo del fragmento de vídeo actual, analizando sintácticamente los vectores de movimiento y otros elementos sintácticos, y usa la información de predicción para producir los bloques predictivos para el bloque de vídeo actual que se está descodificando. Por ejemplo, la unidad de compensación de movimiento 72 usa algunos de los elementos sintácticos recibidos para determinar un modo de predicción (por ejemplo, predicción intra o inter) usado para codificar los bloques de vídeo del fragmento de vídeo, un tipo de fragmento de predicción inter (por ejemplo, un fragmento B, un fragmento P o un fragmento GPB), 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 predicción inter para cada bloque de vídeo intercodificado del fragmento y otra información para descodificar los bloques de vídeo en el fragmento de vídeo actual.

65 [0215] La unidad de compensación de movimiento 72 también puede realizar la interpolación en base a filtros de interpolación. La unidad de compensación de movimiento 72 puede usar filtros de interpolación como se usan por el codificador de vídeo 20 durante la codificación de los bloques de vídeo para calcular valores interpolados para píxeles

fraccionarios de los bloques de referencia. En este caso, la unidad de compensación de movimiento 72 puede determinar los filtros de interpolación usados por el codificador de vídeo 20 de los elementos sintácticos recibidos y usar los filtros de interpolación para producir bloques predictivos.

5 **[0216]** La unidad de cuantificación inversa 76 cuantifica de manera inversa, es decir, descuantifica, los coeficientes de transformada cuantificados proporcionados en el flujo de bits y descodificados por la unidad de descodificación por entropía 70. El procedimiento de cuantización inversa puede incluir el uso de un parámetro de cuantización QPy, calculado por el descodificador de vídeo 30 para cada bloque de vídeo en el fragmento de vídeo, para determinar un grado de cuantización y, del mismo modo, un grado de cuantización inversa que se debería aplicar.

10 **[0217]** La unidad de procesamiento de transformada inversa 78 aplica una transformada inversa, por ejemplo una DCT inversa, una transformada inversa entera o un procedimiento de transformada inversa conceptualmente similar, a los coeficientes de transformada para producir bloques residuales en el dominio de píxeles.

15 **[0218]** Después de que la unidad de compensación de movimiento 72 genera el bloque predictivo para el bloque de vídeo actual en base a los vectores de movimiento y otros elementos sintácticos, el descodificador de vídeo 30 forma un bloque de vídeo descodificado sumando los bloques residuales de la unidad de procesamiento de transformada inversa 78 a los bloques predictivos correspondientes generados por la unidad de compensación de movimiento 72. El sumador 90 representa el componente o componentes que realizan esta operación de suma. La unidad de filtro de bucle 63 puede incluir un filtro de eliminación de bloques y un filtro SAO para filtrar las fronteras del bloque para eliminar distorsiones de pixelado del vídeo reconstruido. También se pueden usar filtros adicionales (en el bucle o tras el bucle), además del filtro de eliminación de bloques. Dichos filtros no se muestran por razones de brevedad, pero, si se desea, pueden filtrar la salida del sumador 80 (como un filtro en el bucle). Los bloques de vídeo descodificados en una trama o imagen dada se almacenan a continuación en la memoria intermedia de imágenes descodificadas 82, que almacena imágenes de referencia usadas para una posterior compensación de movimiento. La memoria intermedia de imágenes descodificadas 82 también almacena vídeo descodificado para su presentación posterior en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la FIG. 1.

30 **[0219]** Como se analizará con más detalle a continuación con referencia a las FIGS. 24-26, el descodificador de vídeo 30 se puede configurar para emplear las técnicas de la presente divulgación descritas anteriormente. En particular, el descodificador de vídeo 30 se puede configurar para particionar las PU en subbloques cuando dichas PU se separan de acuerdo con un modo de particionamiento asimétrica. A continuación, el descodificador de vídeo 30 se puede configurar para heredar y/o derivar vectores de movimiento o vectores de movimiento de disparidad para cada uno de los subbloques.

35 **[0220]** La FIG. 21 es un diagrama de flujo que ilustra un procedimiento de codificación de ejemplo de la divulgación. Las técnicas de la FIG. 21 se pueden implementar por una o más unidades estructurales del codificador de vídeo 20, tal como por la unidad de selección de modo 41, la unidad de particionamiento 48 y/o la unidad de compensación de movimiento 44.

40 **[0221]** En un ejemplo de la divulgación, el codificador de vídeo 20 (por ejemplo, usando la unidad de selección de modo 41 y la unidad de particionamiento 48) se puede configurar para generar un bloque de datos de vídeo usando AMP, en el que el bloque de datos de vídeo se predice unidireccionalmente usando BVSP, y tiene un tamaño de 16x12, 12x16, 16x4 o 4x16 (2100). En un ejemplo de la divulgación, el bloque de datos de vídeo es una unidad de predicción.

45 **[0222]** El codificador de vídeo 20, usando la unidad de particionamiento 48, se puede configurar además para particionar el bloque de datos de vídeo en subbloques, teniendo cada subbloque un tamaño de 8x4 o 4x8 (2110), y derivar (por ejemplo, usando la unidad de compensación de movimiento 44) un respectivo vector de movimiento de disparidad para cada uno de los subbloques de un bloque de profundidad correspondiente en una imagen de profundidad correspondiente a una imagen de referencia (2120). El codificador de vídeo 20 (por ejemplo, usando la unidad de compensación de movimiento 44) se puede configurar además para sintetizar un respectivo bloque de referencia para cada uno de los subbloques usando los respectivos vectores de movimiento de disparidad derivados (2130), y codificar (por ejemplo, usando la unidad de compensación de movimiento 44) el bloque de datos de vídeo realizando una compensación de movimiento en cada uno de los subbloques usando los respectivos bloques de referencia sintetizados (2140).

50 **[0223]** En otro ejemplo de la divulgación, el codificador de vídeo 20 se puede configurar además para generar uno o más elementos sintácticos que indican que la unidad de predicción se codifica usando AMP, y que indican que la unidad de predicción se predice unidireccionalmente usando BVSP, y generar un índice de candidato de fusión que apunta a un candidato de BVSP.

55 **[0224]** En otro ejemplo de la divulgación, el codificador de vídeo 20 (por ejemplo, usando la unidad de compensación de movimiento 44) se puede configurar para derivar el respectivo vector de movimiento de disparidad para cada uno de los subbloques derivando un vector de disparidad para el bloque de datos de vídeo, ubicando el bloque de profundidad correspondiente para cada uno de los subbloques usando el vector de disparidad derivado, y convirtiendo

un valor de profundidad seleccionado del bloque de profundidad correspondiente para cada uno de los subbloques en el vector de movimiento de disparidad respectivo.

5 **[0225]** La FIG. 22 es un diagrama de flujo que ilustra otro procedimiento de codificación de ejemplo de la divulgación. Las técnicas de la FIG. 22 se pueden implementar por una o más unidades estructurales del codificador de vídeo 20, incluyendo la unidad de selección de modo 41, la unidad de particionamiento 48 y/o la unidad de compensación de movimiento 44.

10 **[0226]** En un ejemplo de la divulgación, el codificador de vídeo 20 (por ejemplo, la unidad de selección de modo 41 y la unidad de particionamiento 48) se puede configurar para generar un segundo bloque de datos de vídeo usando AMP, en el que el segundo bloque de datos de vídeo se codifica usando al menos uno de predicción de movimiento entre vistas o MVI, y tiene un tamaño de 16x4 o 4x16 (2200). El codificador de vídeo 20 (por ejemplo, usando la unidad de particionamiento 48) se puede configurar además para particionar el segundo bloque de datos de vídeo en subbloques, teniendo cada subbloque un tamaño de 8x4 o 4x8 (2210), y derivar (por ejemplo, usando la unidad de compensación de movimiento 44) información de movimiento para cada uno de los subbloques de un bloque de referencia respectivo (2220). A continuación, el codificador de vídeo 20 puede codificar el segundo bloque de datos de vídeo realizando una compensación de movimiento en cada uno de los subbloques usando la información de movimiento derivada y una lista de imágenes de referencia (2230).

20 **[0227]** En otro ejemplo de la divulgación, el codificador de vídeo 20 (por ejemplo, usando la unidad de compensación de movimiento 44) se puede configurar para realizar la compensación de movimiento realizando una compensación de movimiento unidireccional con respecto a una imagen en la una lista de imágenes de referencia.

25 **[0228]** La FIG. 23 es un diagrama de flujo que ilustra otro procedimiento de codificación de ejemplo de la divulgación. Las técnicas de la FIG. 23 se pueden implementar por una o más unidades estructurales del codificador de vídeo 20, tal como por la unidad de selección de modo 41, la unidad de particionamiento 48 y/o la unidad de compensación de movimiento 44.

30 **[0229]** En un ejemplo de la divulgación, el codificador de vídeo 20 se puede configurar para generar 20 (por ejemplo, usando la unidad de selección de modo 41 y la unidad de particionamiento 48) un segundo bloque de datos de vídeo usando AMP, en el que el segundo bloque de datos de vídeo se codifica usando al menos uno de predicción de movimiento entre vistas o herencia del vector de movimiento, y tiene un tamaño de 16x12 o 12x16 (2300), particionar (por ejemplo, usando la unidad de particionamiento 48) el segundo bloque de datos de vídeo en múltiples subbloques (2310), y codificar (por ejemplo, usando la unidad de compensación de movimiento 44) cada uno de los múltiples subbloques con predicción unipredictiva (2320).

40 **[0230]** La FIG. 24 es un diagrama de flujo que ilustra un procedimiento de descodificación de ejemplo de la divulgación. Las técnicas de la FIG. 24 se pueden implementar por una o más unidades estructurales del descodificador de vídeo 30, tal como por la unidad de compensación de movimiento 72.

45 **[0231]** En un ejemplo de la divulgación, el descodificador de vídeo 30 se puede configurar para recibir datos residuales correspondientes a un bloque de datos de vídeo, en el que el bloque de datos de vídeo se codifica usando AMP, se predice unidireccionalmente usando BVSP, y tiene un tamaño de 16x12, 12x16, 16x4 o 4x16 (2400). En un ejemplo de la divulgación, el bloque de datos de vídeo es una unidad de predicción. El descodificador de vídeo 30 se puede configurar además para particionar el bloque de datos de vídeo en subbloques, teniendo cada subbloque un tamaño de 8x4 o 4x8 (2410), y derivar un respectivo vector de movimiento de disparidad para cada uno de los subbloques de un bloque de profundidad correspondiente en una imagen de profundidad correspondiente a una imagen de referencia (2420).

50 **[0232]** El descodificador de vídeo 30 se puede configurar además para sintetizar un respectivo bloque de referencia para cada uno de los subbloques usando los respectivos vectores de movimiento de disparidad derivados (2430), y descodificar el bloque de datos de vídeo realizando la compensación de movimiento en cada uno de los subbloques usando los datos residuales y los respectivos bloques de referencia sintetizados (2440).

55 **[0233]** En otro ejemplo de la divulgación, el descodificador de vídeo 30 se puede configurar además para recibir uno o más elementos sintácticos que indican que la unidad de predicción se codifica usando particionamiento de movimiento asimétrica, y que indican que la unidad de predicción se predice unidireccionalmente usando predicción de síntesis de vista inversa, y recibir un índice de candidato de fusión que apunta a un candidato de BVSP.

60 **[0234]** En otro ejemplo de la divulgación, el descodificador de vídeo 30 se puede configurar además para derivar el vector de movimiento de disparidad respectivo para cada uno de los subbloques derivando un vector de disparidad para el bloque de datos de vídeo, ubicando el bloque de profundidad correspondiente para cada uno de los subbloques usando el vector de disparidad derivado, y convirtiendo un valor de profundidad seleccionado del bloque de profundidad correspondiente para cada uno de los subbloques en el vector de movimiento de disparidad respectivo.

65

[0235] La FIG. 25 es un diagrama de flujo que ilustra un procedimiento de descodificación de ejemplo de la divulgación. Las técnicas de la FIG. 23 se pueden implementar por una o más unidades estructurales del descodificador de vídeo 30, tal como por la unidad de compensación de movimiento 72.

5 **[0236]** En un ejemplo de la divulgación, el descodificador de vídeo 30 se puede configurar para recibir datos residuales correspondientes a un segundo bloque de datos de vídeo, en el que el segundo bloque de datos de vídeo se codifica usando al menos uno de predicción de movimiento entre vistas o MVI, y tiene un tamaño de 16x4 o 4x16 (2500), dividir el segundo bloque de datos de vídeo en subbloques, teniendo cada subbloque un tamaño de 8x4 o 4x8 (2510), derivar información de movimiento para cada uno de los subbloques de un respectivo bloque de referencia (2520), y
10 descodificar el segundo bloque de datos de vídeo realizando la compensación de movimiento en cada uno de los subbloques usando los datos residuales, la información de movimiento derivada y una lista de imágenes de referencia.

[0237] En otro ejemplo de la divulgación, el descodificador de vídeo 30 se puede configurar además para realizar la compensación de movimiento realizando una compensación de movimiento unidireccional con respecto a una imagen en la una lista de imágenes de referencia.
15

[0238] La FIG. 26 es un diagrama de flujo que ilustra un procedimiento de descodificación de ejemplo de la divulgación. Las técnicas de la FIG. 23 se pueden implementar por una o más unidades estructurales del descodificador de vídeo 30, incluyendo la unidad de compensación de movimiento 72.
20

[0239] En un ejemplo de la divulgación, el descodificador de vídeo 30 se puede configurar además para recibir datos residuales correspondientes a un segundo bloque de datos de vídeo, en el que el segundo bloque de datos de vídeo se codifica usando al menos uno de predicción de movimiento entre vistas o MVI, y tiene un tamaño de 16x12 o 12x16 (2600), particionar el segundo bloque de datos de vídeo en múltiples subbloques (2610), y descodificar cada uno de los múltiples subbloques con predicción unipredictiva.
25

[0240] Como se explica anteriormente, las técnicas de la presente divulgación incluyen técnicas de codificación y descodificación de vídeo cuando se aplica AMP, BVSP, predicción de movimiento entre vistas y/o MVI en bloques de datos de vídeo. En particular, las técnicas de la presente divulgación proporcionan una codificación más exacta al dirigir las técnicas de codificación a subbloques de las PU separados con AMP. Por ejemplo, obtener vectores de movimiento de disparidad separados para subbloques de una PU separada con AMP cuando dicha PU se codifica usando BVSP puede incrementar la exactitud de la predicción de síntesis de vista y de movimiento, y por tanto, la eficacia de codificación. Como otro ejemplo, obtener información de movimiento separada para subbloques de una PU separada con AMP cuando dicha PU se codifica usando predicción de movimiento entre vistas y/o MVI puede incrementar la exactitud de la predicción de movimiento y, por tanto, la eficacia de codificación.
30
35

[0241] Se debe reconocer que, dependiendo del ejemplo, determinados actos o sucesos de cualquiera de las técnicas descritas en el presente documento, se pueden realizar en una secuencia diferente, se pueden añadir, fusionar u omitir por completo (por ejemplo, no todos los actos o sucesos descritos son necesarios para la puesta en práctica de las técnicas). Además, en determinados ejemplos, los actos o sucesos se pueden realizar simultáneamente, por ejemplo, a través de procesamiento de múltiples subprocedimientos, procesamiento de interrupciones o múltiples procesadores, en lugar de secuencialmente.
40

[0242] En uno o más ejemplos, las funciones descritas se pueden implementar en hardware, software, firmware o cualquier combinación de los mismos. Si se implementan en software, las funciones se pueden almacenar en, o transmitir sobre, un medio legible por ordenador, como una o más instrucciones o código, y ejecutar por una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible tal como medios de almacenamiento de datos, o medios de comunicación que incluyen cualquier medio que facilita la transferencia de un programa informático de un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador pueden corresponder, en general, a (1) medios de almacenamiento tangibles legibles por ordenador que son no transitorios o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que se pueda acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en la presente divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.
45
50
55

[0243] A modo de ejemplo, y no de limitación, dichos medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento de disco óptico, almacenamiento de disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que se pueda usar para almacenar un código de programa deseado en forma de instrucciones o estructuras de datos y al que se pueda acceder por un ordenador. Además, cualquier conexión recibe apropiadamente la denominación de medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, 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 están incluidos en la
60
65

definición de medio. Sin embargo, se deberá entender que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que, en cambio, están dirigidos a medios de almacenamiento no transitorio tangibles. Los discos, como se usan en el presente documento, incluyen el disco compacto (CD), el disco láser, el disco óptico, el disco versátil digital (DVD), el disco flexible y el disco Blu-ray, donde algunos discos reproducen normalmente los datos magnéticamente, mientras que otros discos reproducen los datos ópticamente con láseres. Las combinaciones de los anteriores también se deben incluir dentro del alcance de los medios legibles por ordenador.

[0244] Las instrucciones se pueden ejecutar por uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), matrices lógicas programables *in situ* (FPGA) u otros circuitos lógicos discretos o integrados equivalentes. En consecuencia, el término "procesador", como se usa en el presente documento, se puede referir a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento se puede proporcionar en módulos de hardware y/o software dedicados configurados para la codificación y la decodificación, o incorporarse en un códec combinado. Además, las técnicas se podrían implementar por completo en uno o más circuitos o elementos lógicos.

[0245] Las técnicas de la presente divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluyendo un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En la presente divulgación se describen diversos componentes, módulos o unidades para destacar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no se requiere necesariamente su realización por diferentes unidades de hardware. En su lugar, como se describe anteriormente, diversas unidades se pueden combinar en una unidad de hardware de códec o proporcionar por un conjunto de unidades de hardware interoperativas, incluyendo uno o más procesadores como se describe anteriormente, junto con software y/o firmware adecuado.

[0246] Se han descrito diversos ejemplos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.

REIVINDICACIONES

1. Un procedimiento de descodificación de datos de vídeo, comprendiendo el procedimiento:
 - 5 recibir datos residuales correspondientes a un bloque de datos de vídeo, en el que el bloque de datos de vídeo se ha codificado usando particionamiento de movimiento asimétrica, se predice unidireccionalmente usando la predicción de síntesis de vista inversa, BVSP, y tiene un tamaño de 16x12, 12x16, 16x4 o 4x16, y en el que el bloque de datos de vídeo es una unidad de predicción de una unidad de codificación;
 - 10 particionar el bloque de datos de vídeo en subbloques, teniendo cada subbloque un tamaño de 8x4 o 4x8;
 - derivar un respectivo vector de movimiento de disparidad para cada uno de los subbloques de un bloque de profundidad correspondiente en una imagen de profundidad correspondiente a una imagen de referencia;
 - 15 sintetizar un respectivo bloque de referencia para cada uno de los subbloques usando los respectivos vectores de movimiento de disparidad derivados; y
 - descodificar el bloque de datos de vídeo realizando la compensación de movimiento en cada uno de los subbloques usando los datos residuales y los respectivos bloques de referencia sintetizados.
 - 20
2. El procedimiento de la reivindicación 1, que comprende además:
 - 25 recibir uno o más elementos sintácticos que indican que la unidad de predicción se ha codificado usando particionamiento de movimiento asimétrica, y que indican que la unidad de predicción se predice unidireccionalmente usando predicción de síntesis de vista inversa; y
 - recibir un índice de candidato de fusión que apunta a un candidato de BVSP.
3. El procedimiento de la reivindicación 1, en el que derivar el vector de movimiento de disparidad respectivo para cada uno de los subbloques comprende:
 - 30 derivar un vector de disparidad para el bloque de datos de vídeo;
 - ubicar el bloque de profundidad correspondiente para cada uno de los subbloques usando el vector de disparidad derivado; y
 - 35 convertir un valor de profundidad seleccionado del bloque de profundidad correspondiente para cada uno de los subbloques en el respectivo vector de movimiento de disparidad.
4. El procedimiento de la reivindicación 1, en el que el bloque de datos de vídeo es un primer bloque de datos de vídeo, comprendiendo además el procedimiento:
 - 40 recibir datos residuales correspondientes a un segundo bloque de datos de vídeo, en el que el segundo bloque de datos de vídeo se ha codificado usando al menos uno de predicción de movimiento entre vistas o herencia del vector de movimiento, y tiene un tamaño de 16x4 o 4x16;
 - 45 particionar el segundo bloque de datos de vídeo en subbloques, teniendo cada subbloque un tamaño de 8x4 o 4x8;
 - 50 derivar información de movimiento para cada uno de los subbloques de un respectivo bloque de referencia; y
 - descodificar el segundo bloque de datos de vídeo realizando una compensación de movimiento en cada uno de los subbloques usando los datos residuales, la información de movimiento derivada y una lista de imágenes de referencia.
 - 55
5. El procedimiento de la reivindicación 4, en el que realizar una compensación de movimiento comprende realizar una compensación de movimiento unidireccional con respecto a una imagen en la una lista de imágenes de referencia.
6. El procedimiento de la reivindicación 1, en el que el bloque de datos de vídeo es un primer bloque de datos de vídeo, comprendiendo además el procedimiento:
 - 60 recibir datos residuales correspondientes a un segundo bloque de datos de vídeo, en el que el segundo bloque de datos de vídeo se ha codificado usando al menos uno de predicción de movimiento entre vistas o herencia del vector de movimiento, y tiene un tamaño de 16x12 o 12x16;
 - 65

particionar el segundo bloque de datos de vídeo en múltiples subbloques; y

descodificar cada uno de los múltiples subbloques con predicción unipredictiva.

- 5
7. Un procedimiento de codificación de datos de vídeo, comprendiendo el procedimiento:
- 10 generar un bloque de datos de vídeo usando particionamiento de movimiento asimétrica, en el que el bloque de datos de vídeo se predice unidireccionalmente usando la predicción de síntesis de vista inversa, BVSP, y tiene un tamaño de 16x12, 12x16, 16x4 o 4x16, y en el que el bloque de datos de vídeo es una unidad de predicción de una unidad de codificación;
- 15 particionar el bloque de datos de vídeo en subbloques, teniendo cada subbloque un tamaño de 8x4 o 4x8;
- 15 derivar un respectivo vector de movimiento de disparidad para cada uno de los subbloques de un bloque de profundidad correspondiente en una imagen de profundidad correspondiente a una imagen de referencia;
- 20 sintetizar un respectivo bloque de referencia para cada uno de los subbloques usando los respectivos vectores de movimiento de disparidad derivados; y
- 20 codificar el bloque de datos de vídeo realizando la compensación de movimiento en cada uno de los subbloques usando los respectivos bloques de referencia sintetizados.
- 25 8. El procedimiento de la reivindicación 7, que comprende además:
- 25 generar uno o más elementos sintácticos que indican que la unidad de predicción se ha codificado usando particionamiento de movimiento asimétrica, y que indican que la unidad de predicción se predice unidireccionalmente usando predicción de síntesis de vista inversa; y
- 30 generar un índice de candidato de fusión que apunta a un candidato de BVSP.
9. El procedimiento de la reivindicación 8, en el que derivar el vector de movimiento de disparidad respectivo para cada uno de los subbloques comprende:
- 35 derivar un vector de disparidad para el bloque de datos de vídeo;
- 35 ubicar el bloque de profundidad correspondiente para cada uno de los subbloques usando el vector de disparidad derivado; y
- 40 convertir un valor de profundidad seleccionado del bloque de profundidad correspondiente para cada uno de los subbloques en el respectivo vector de movimiento de disparidad.
- 45 10. El procedimiento de la reivindicación 7, en el que el bloque de datos de vídeo es un primer bloque de datos de vídeo, comprendiendo además el procedimiento:
- 45 generar un segundo bloque de datos de vídeo usando una particionamiento de movimiento asimétrica, en el que el segundo bloque de datos de vídeo se ha codificado usando al menos uno de predicción de movimiento entre vistas o herencia del vector de movimiento, y tiene un tamaño de 16x4 o 4x16;
- 50 particionar el segundo bloque de datos de vídeo en subbloques, teniendo cada subbloque un tamaño de 8x4 o 4x8;
- 50 derivar información de movimiento para cada uno de los subbloques de un respectivo bloque de referencia; y
- 55 codificar el segundo bloque de datos de vídeo realizando una compensación de movimiento en cada uno de los subbloques usando la información de movimiento derivada y una lista de imágenes de referencia.
- 60 11. El procedimiento de la reivindicación 10, en el que realizar una compensación de movimiento comprende realizar una compensación de movimiento unidireccional con respecto a una imagen en la una lista de imágenes de referencia.
- 65 12. Un aparato configurado para descodificar datos de vídeo, comprendiendo el aparato:
- 65 una memoria de vídeo configurada para almacenar información correspondiente a un bloque de datos de vídeo; y

uno o más procesadores configurados para:

5 recibir datos residuales correspondientes al bloque de datos de vídeo, en el que el bloque de datos de vídeo se ha codificado usando particionamiento de movimiento asimétrica, se predice unidireccionalmente usando la predicción de síntesis de vista inversa, BVSP, y tiene un tamaño de 16x12, 12x16, 16x4 o 4x16, y en el que el bloque de datos de vídeo es una unidad de predicción de una unidad de codificación;

10 particionar el bloque de datos de vídeo en subbloques, teniendo cada subbloque un tamaño de 8x4 o 4x8;

derivar un respectivo vector de movimiento de disparidad para cada uno de los subbloques de un subbloque correspondiente de un bloque de profundidad en una imagen de profundidad correspondiente a una imagen de referencia;

15 sintetizar un respectivo bloque de referencia para cada uno de los subbloques usando los respectivos vectores de movimiento de disparidad derivados; y

20 descodificar el bloque de datos de vídeo realizando la compensación de movimiento en cada uno de los subbloques usando los datos residuales y los respectivos bloques de referencia sintetizados.

13. El aparato de la reivindicación 12, en el que el uno o más procesadores se configuran además para:

25 recibir uno o más elementos sintácticos que indican que la unidad de predicción se ha codificado usando particionamiento de movimiento asimétrica, y que indican que la unidad de predicción se predice unidireccionalmente usando predicción de síntesis de vista inversa; y

recibir un índice de candidato de fusión que apunta a un candidato de BVSP.

14. El aparato de la reivindicación 12, en el que el uno o más procesadores se configuran además para:

30 derivar un vector de disparidad para el bloque de datos de vídeo;

ubicar el bloque de profundidad correspondiente para cada uno de los subbloques usando el vector de disparidad derivado; y

35 convertir un valor de profundidad seleccionado del bloque de profundidad correspondiente para cada uno de los subbloques en el respectivo vector de movimiento de disparidad.

15. El aparato de la reivindicación 12, en el que el bloque de datos de vídeo es un primer bloque de datos de vídeo, y en el que el uno o más procesadores se configuran además para:

40 recibir datos residuales correspondientes a un segundo bloque de datos de vídeo, en el que el segundo bloque de datos de vídeo se ha codificado usando al menos uno de predicción de movimiento entre vistas o herencia del vector de movimiento, y tiene un tamaño de 16x4 o 4x16;

45 particionar el segundo bloque de datos de vídeo en subbloques, teniendo cada subbloque un tamaño de 8x4 o 4x8;

50 derivar información de movimiento para cada uno de los subbloques de un respectivo bloque de referencia; y

descodificar el segundo bloque de datos de vídeo realizando una compensación de movimiento en cada uno de los subbloques usando los datos residuales, la información de movimiento derivada y una lista de imágenes de referencia.

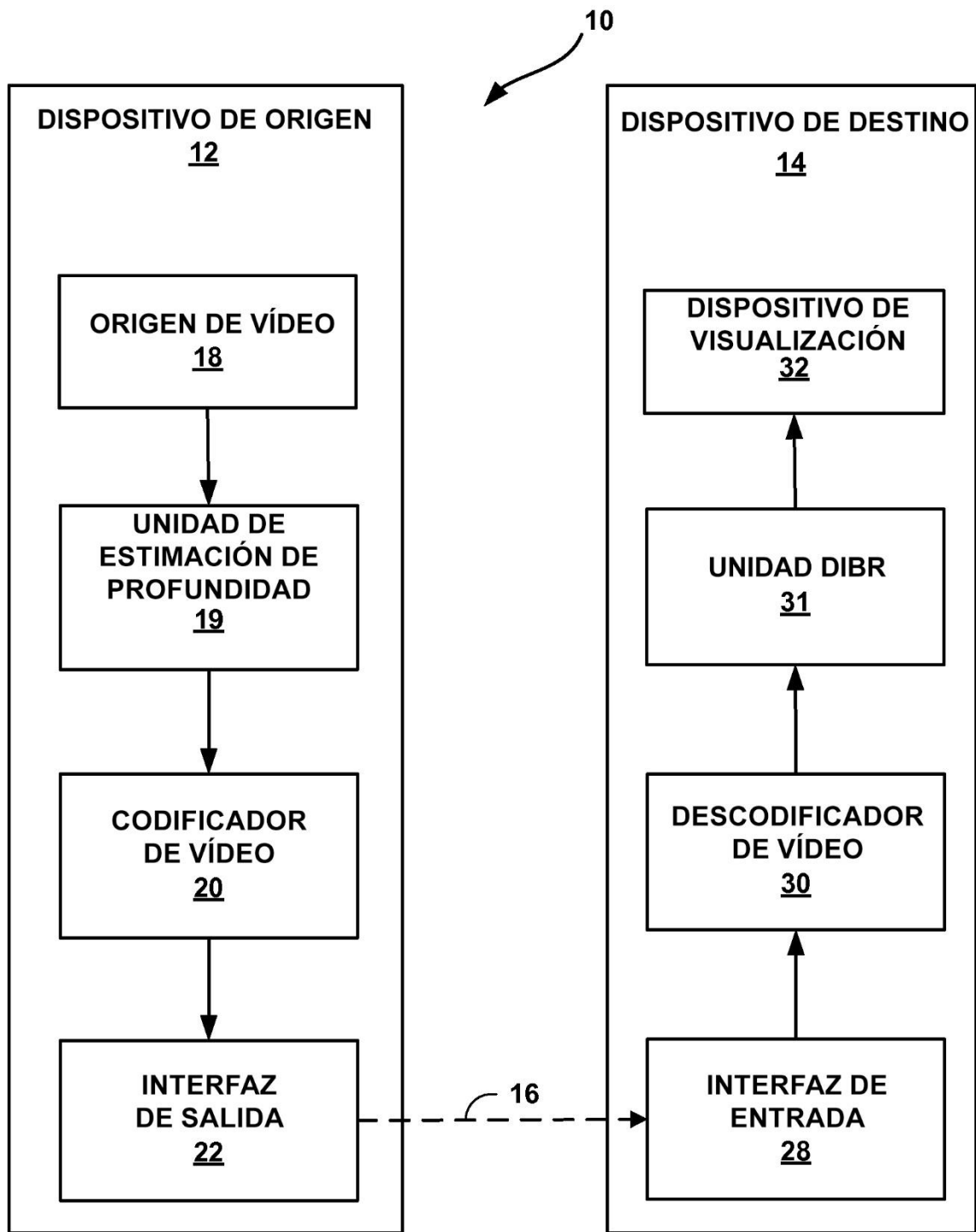


FIG. 1

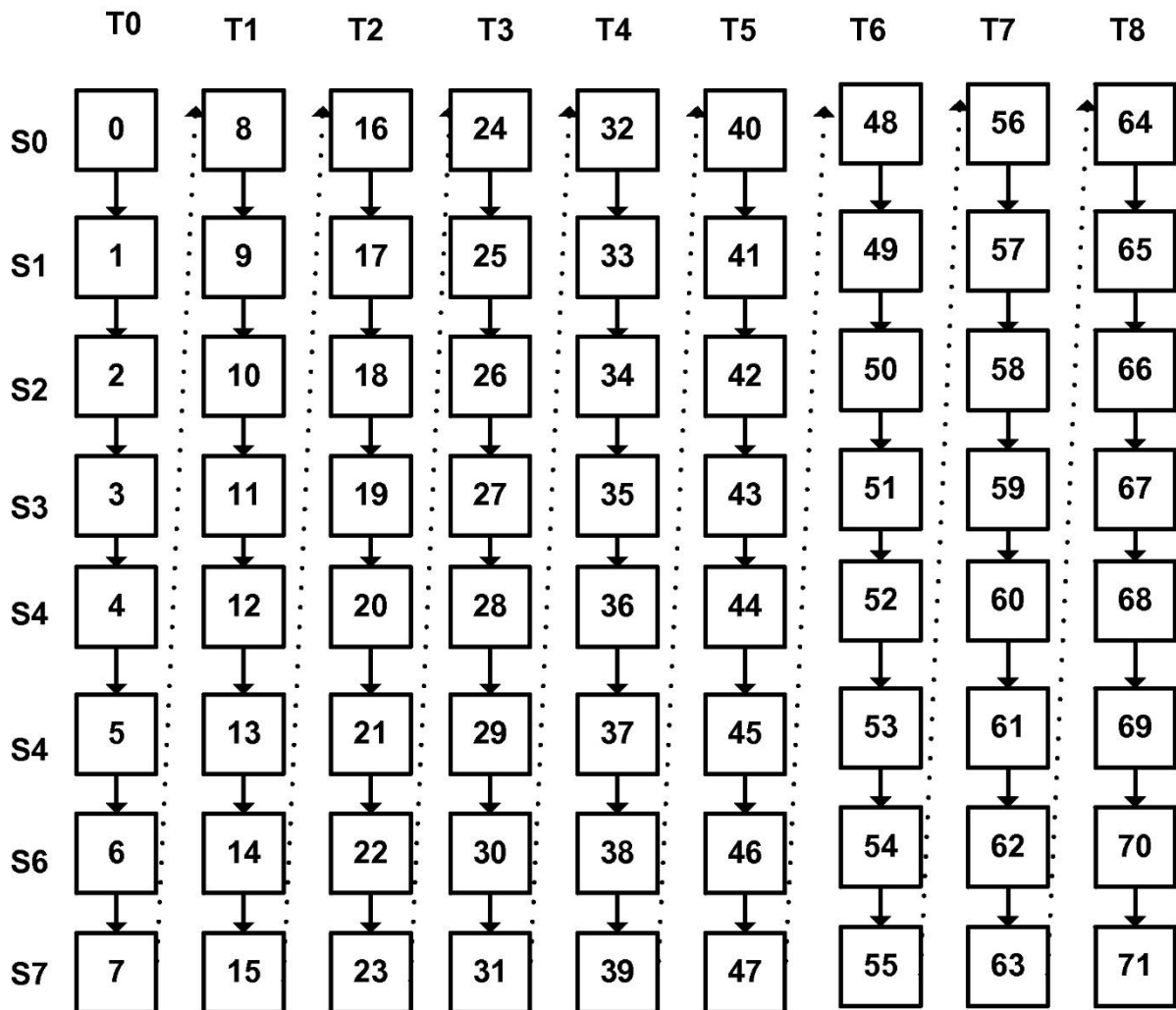


FIG. 2

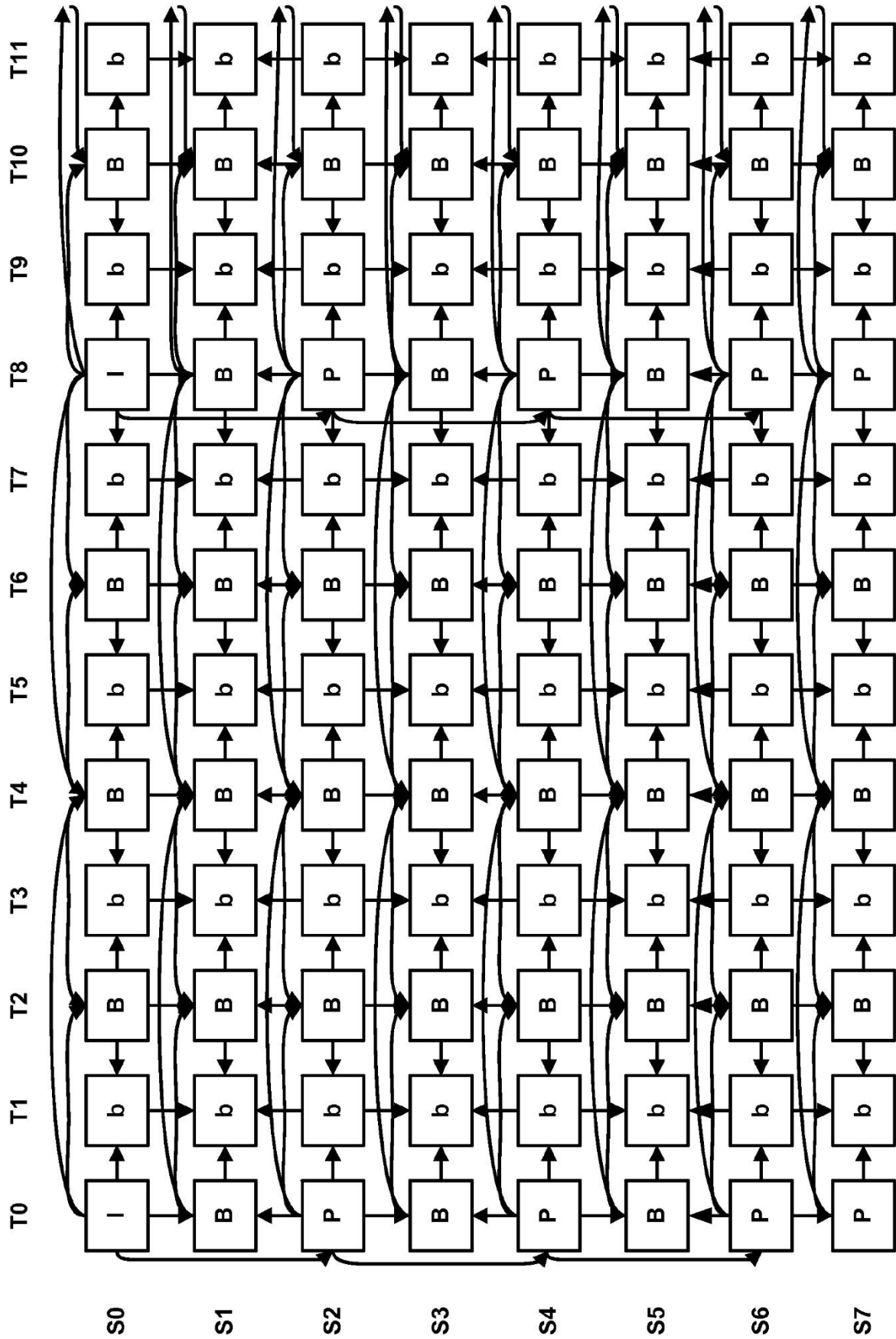


FIG. 3

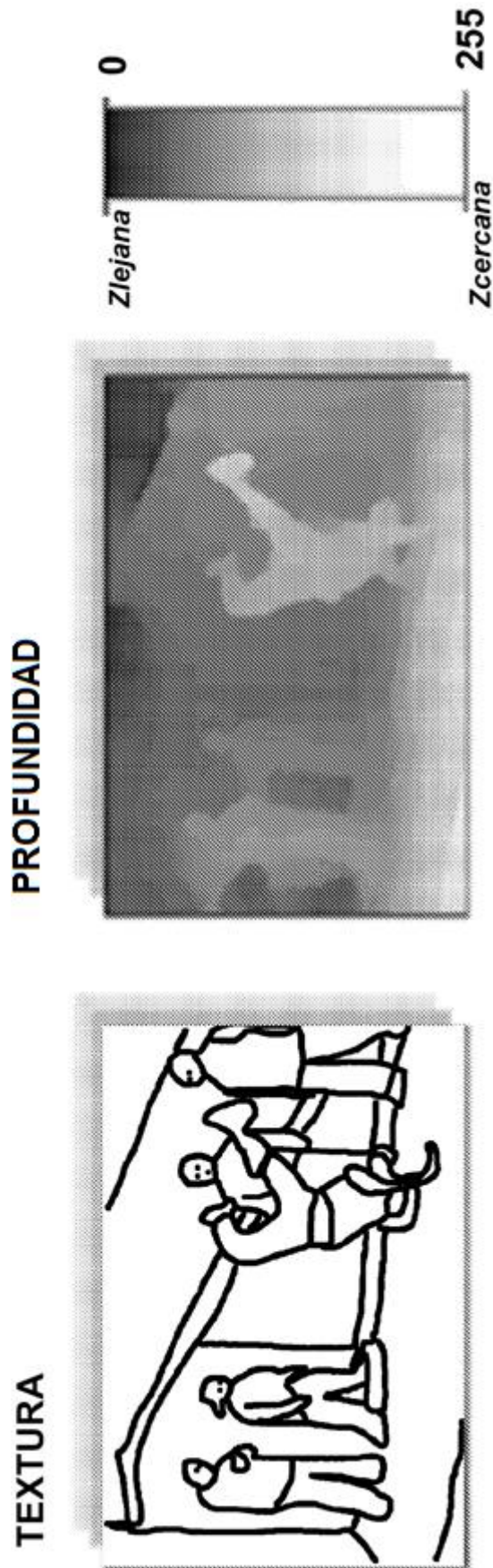


FIG. 4

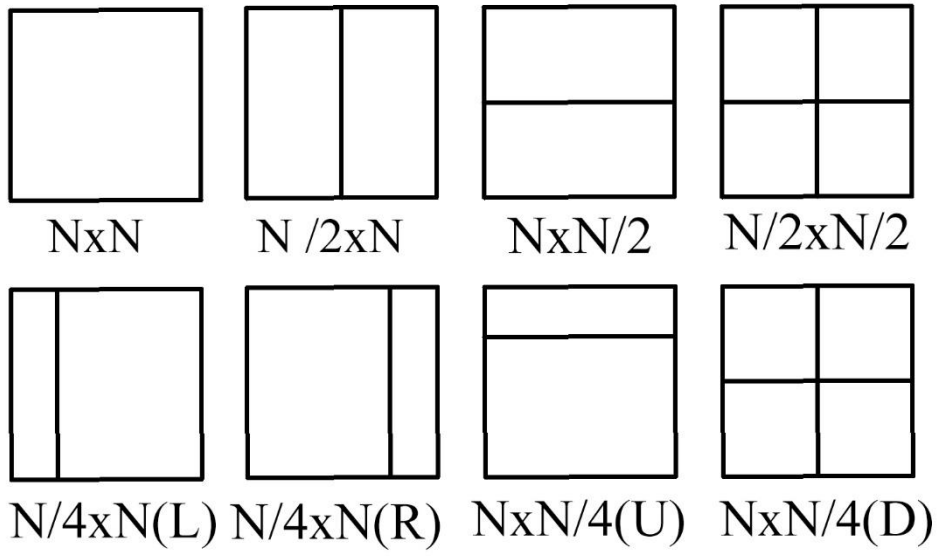


FIG. 5

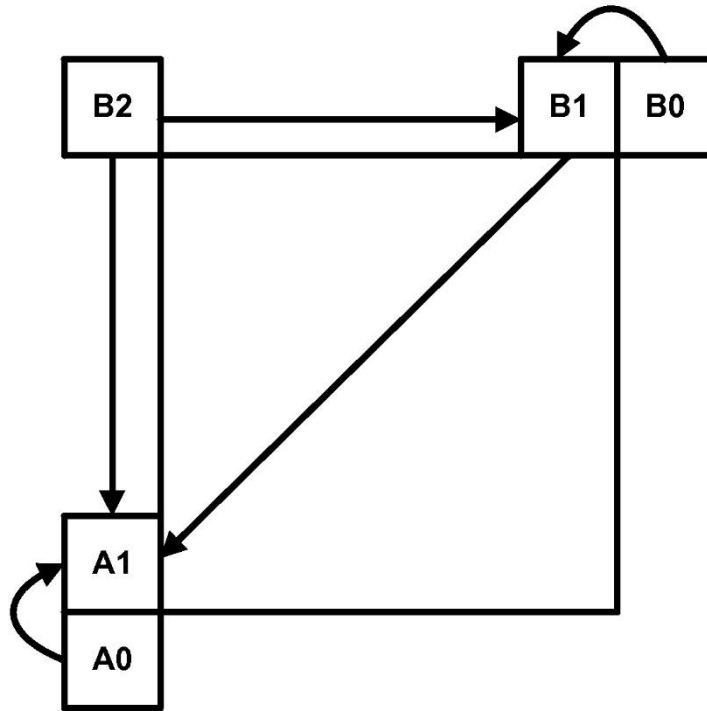


FIG. 6

| | | | | | | | | | | | | |
|-----------|---|---|---|---|---|---|---|---|---|---|----|----|
| comblidx | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| l0CandIdx | 0 | 1 | 0 | 2 | 1 | 2 | 0 | 3 | 1 | 3 | 2 | 3 |
| l1CandIdx | 1 | 0 | 2 | 0 | 2 | 1 | 3 | 0 | 3 | 1 | 3 | 2 |

FIG. 7

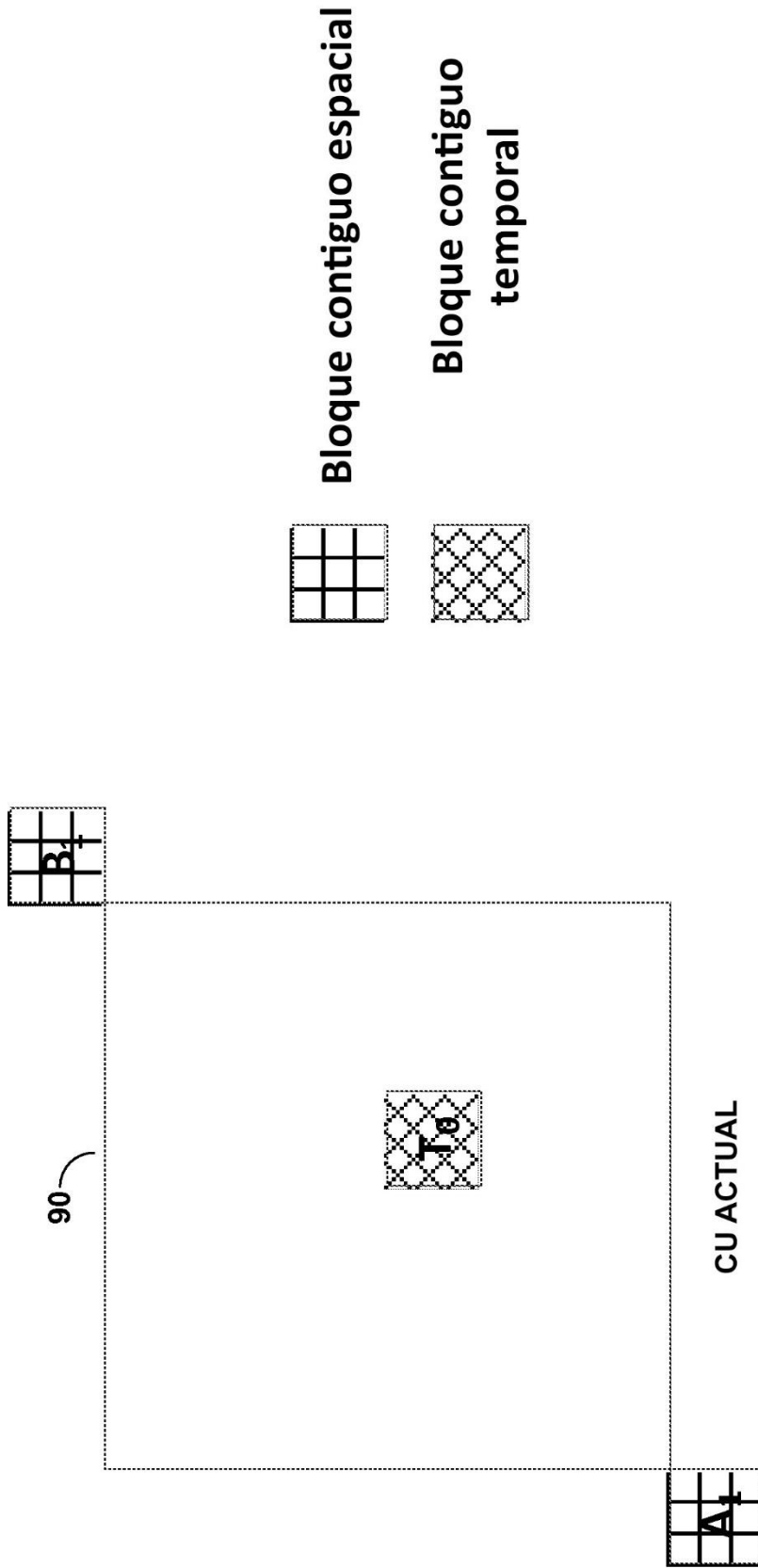


FIG. 8 BLOQUES CONTIGUOS ESPACIALES Y TEMPORALES CON RESPECTO A LA UNIDAD DE CODIFICACIÓN ACTUAL

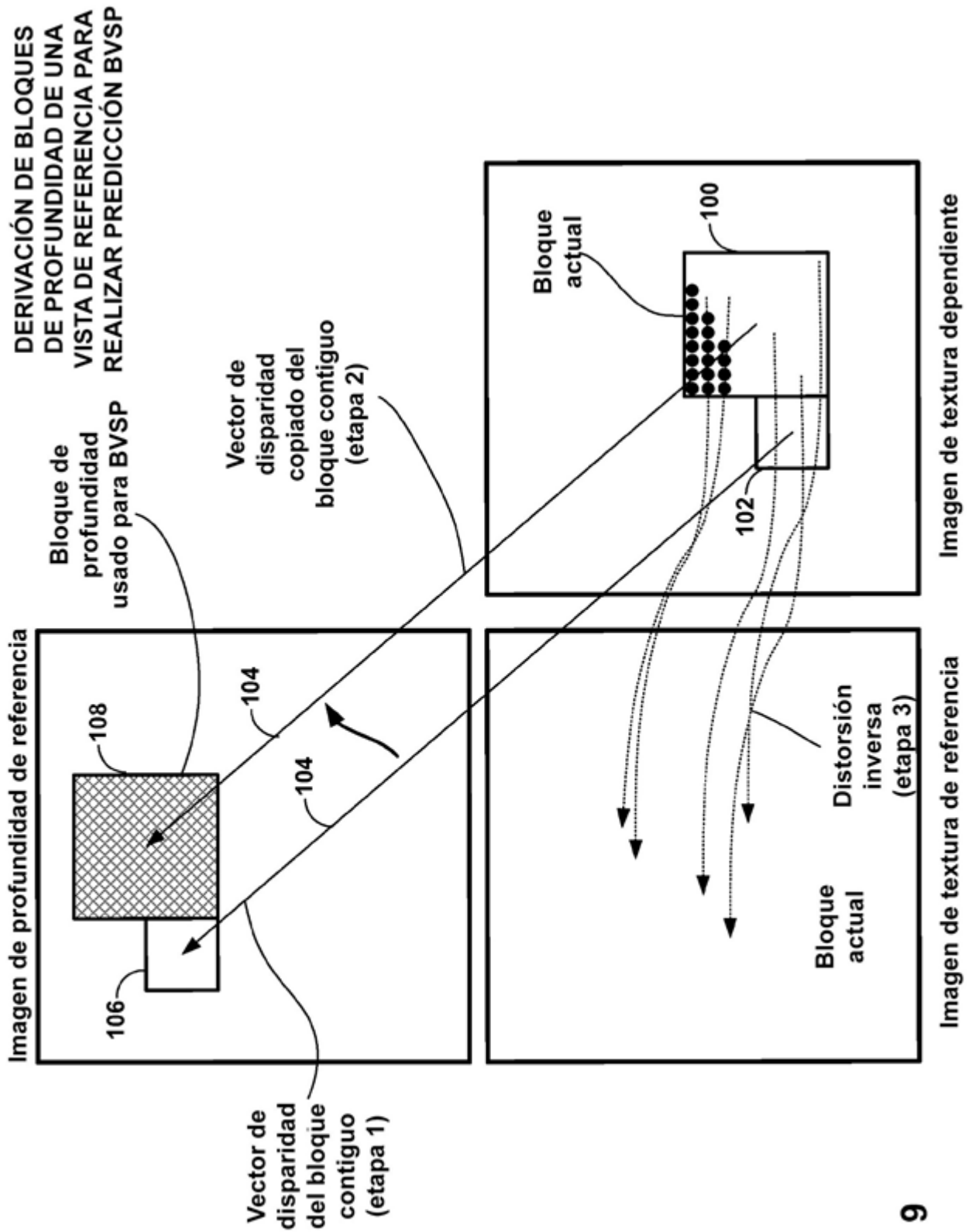
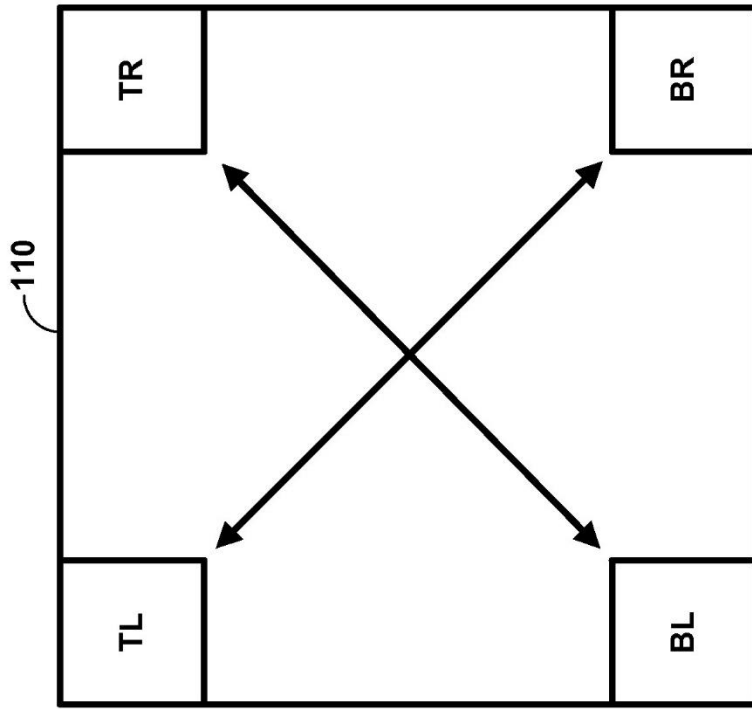


FIG. 9



CUATRO PÍXELES DE ESQUINA DE UN BLOQUE DE PROFUNDIDAD 8X8

FIG. 10

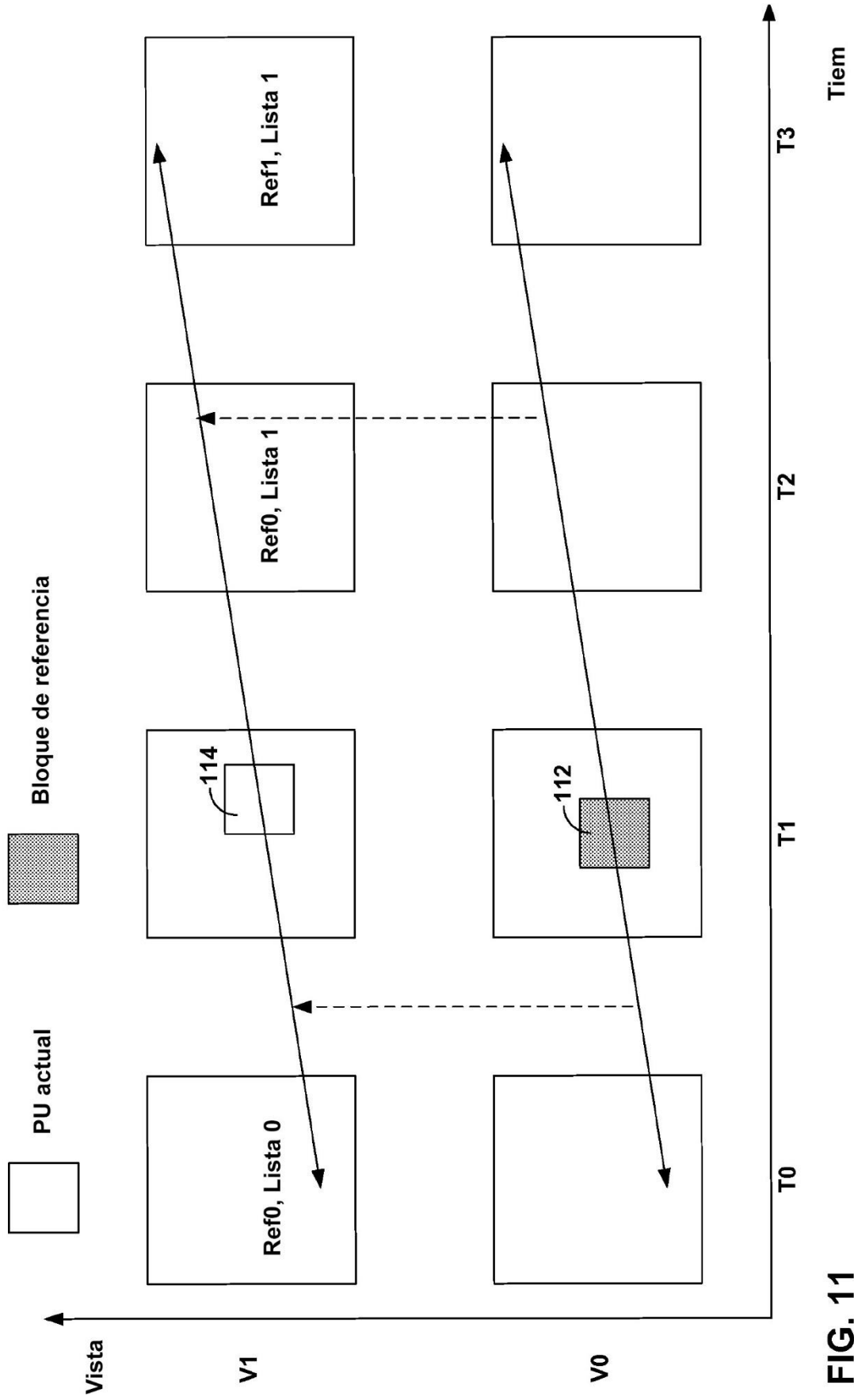


FIG. 11

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

FIG. 12

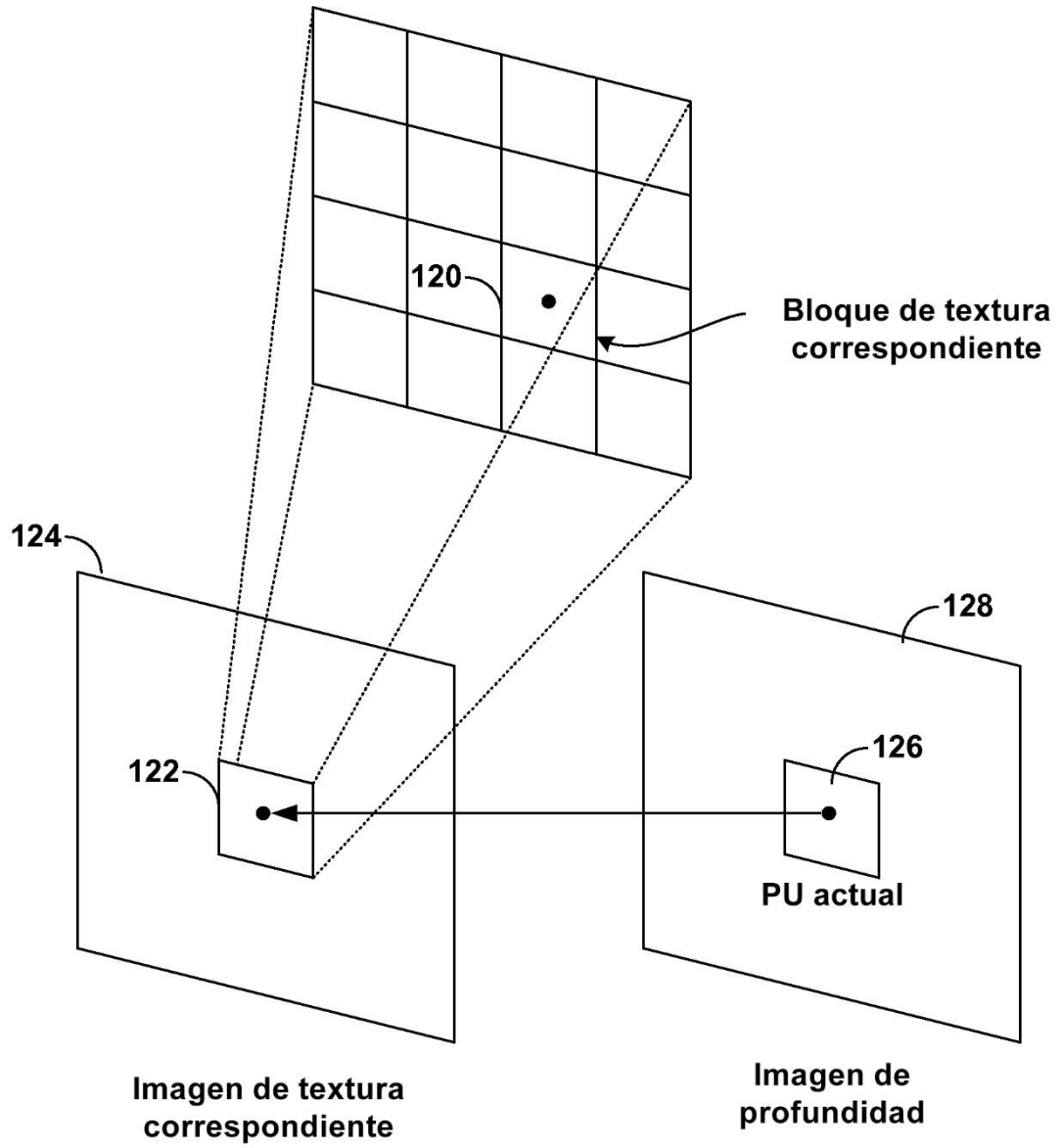


FIG. 13

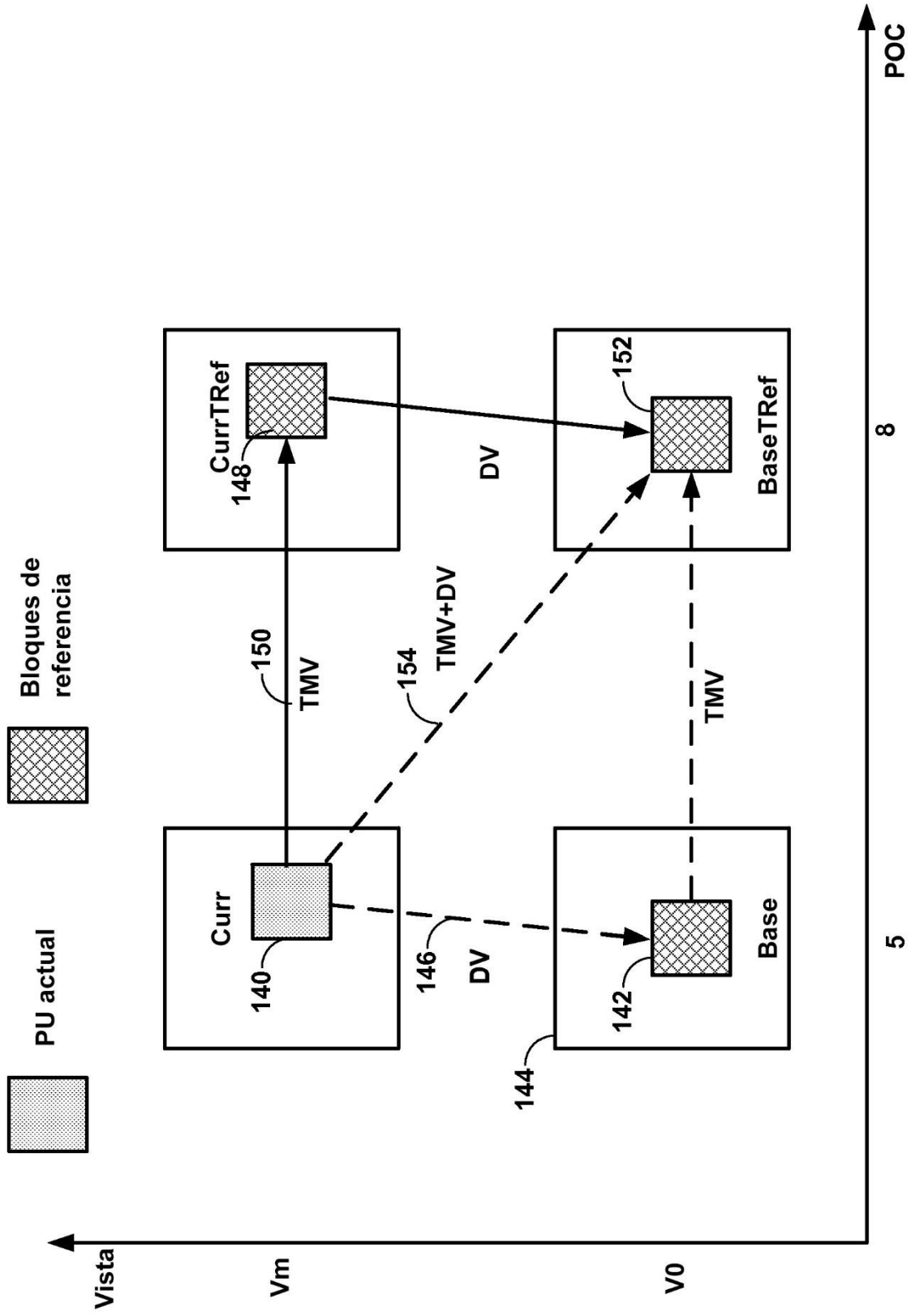


FIG. 14

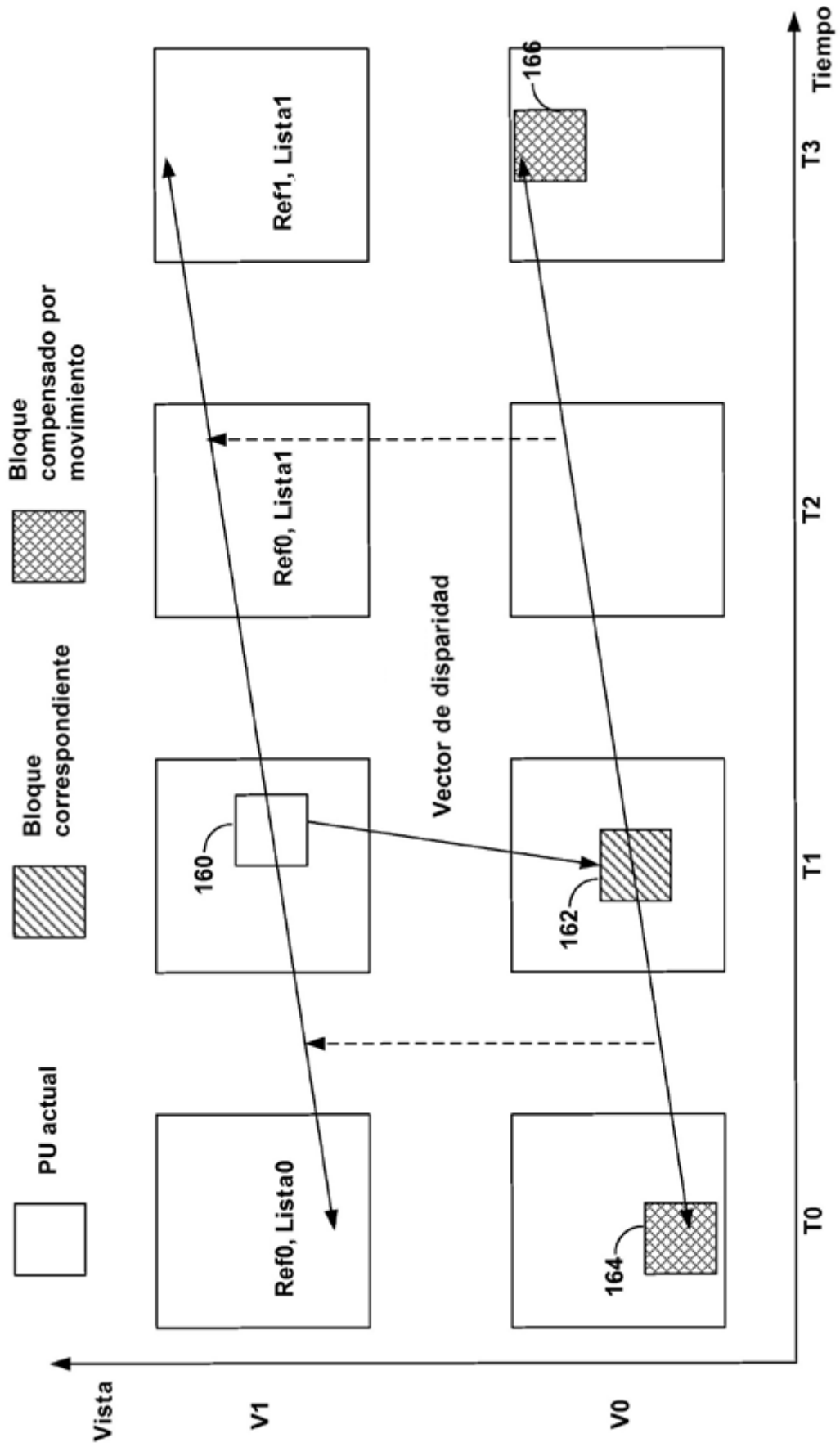


FIG. 15

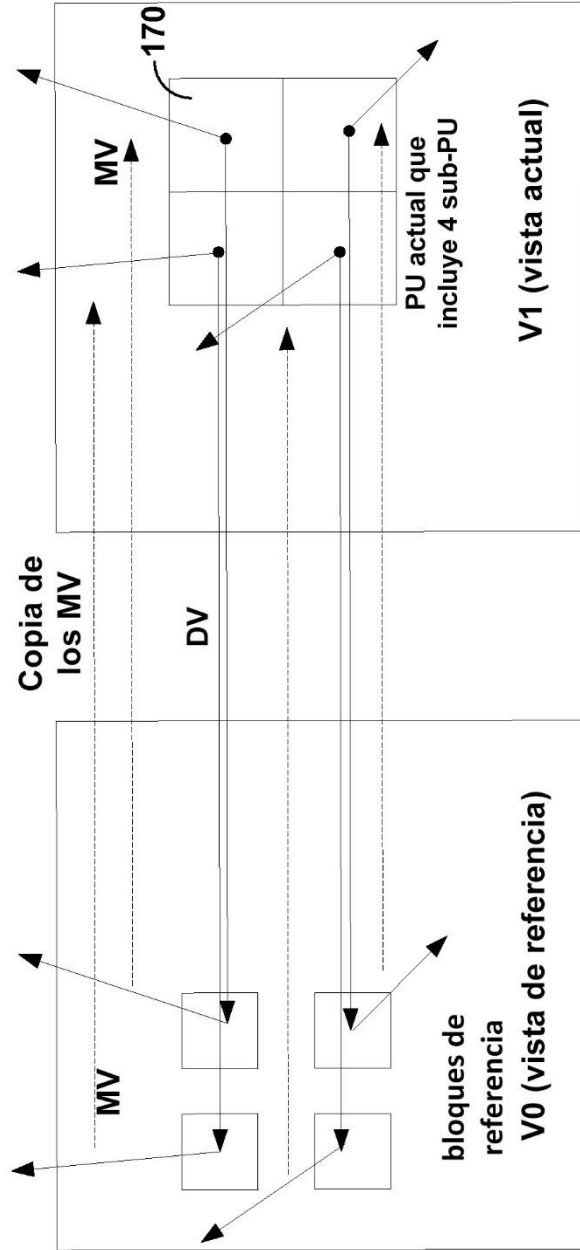


FIG. 16

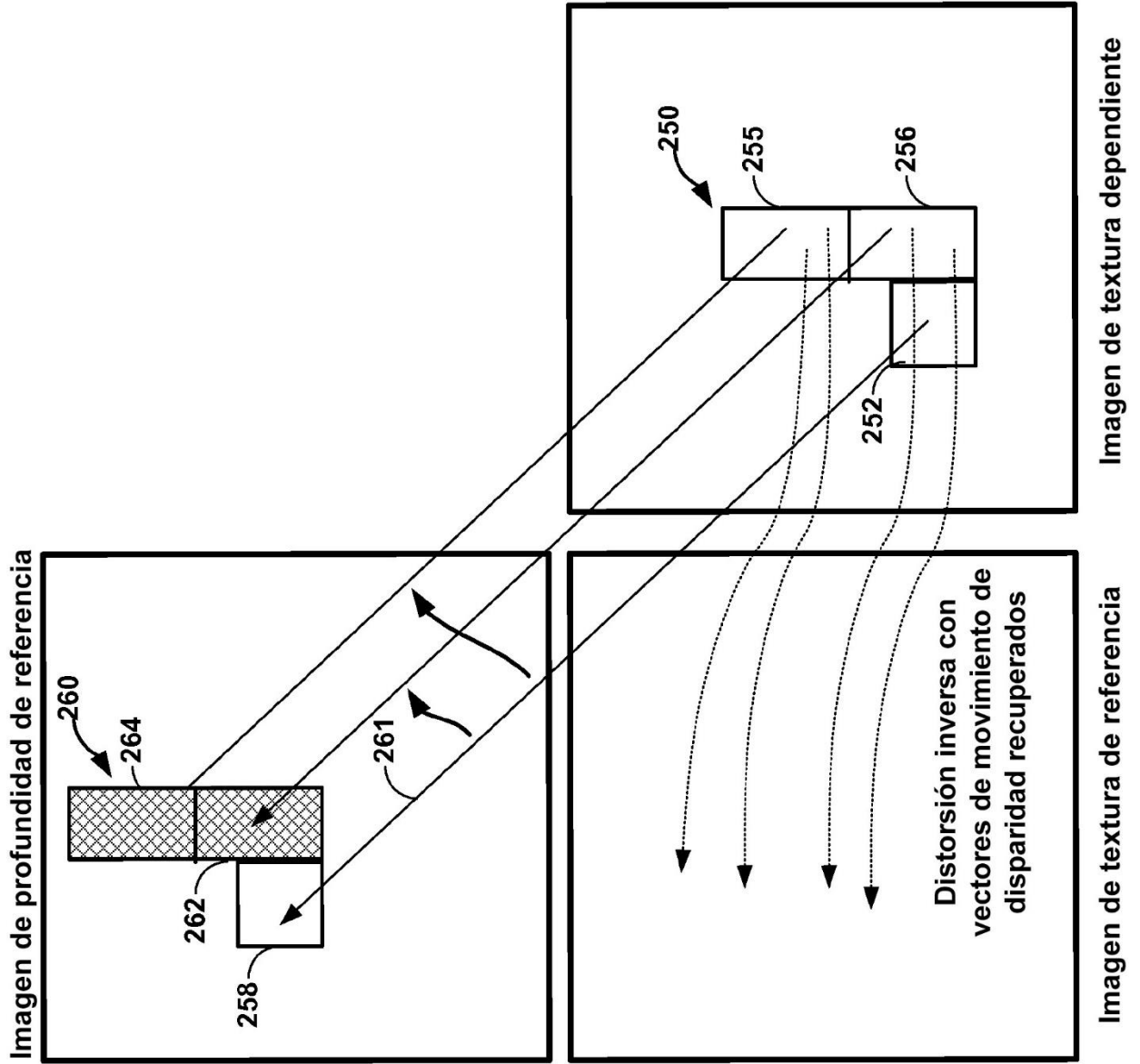


FIG. 17

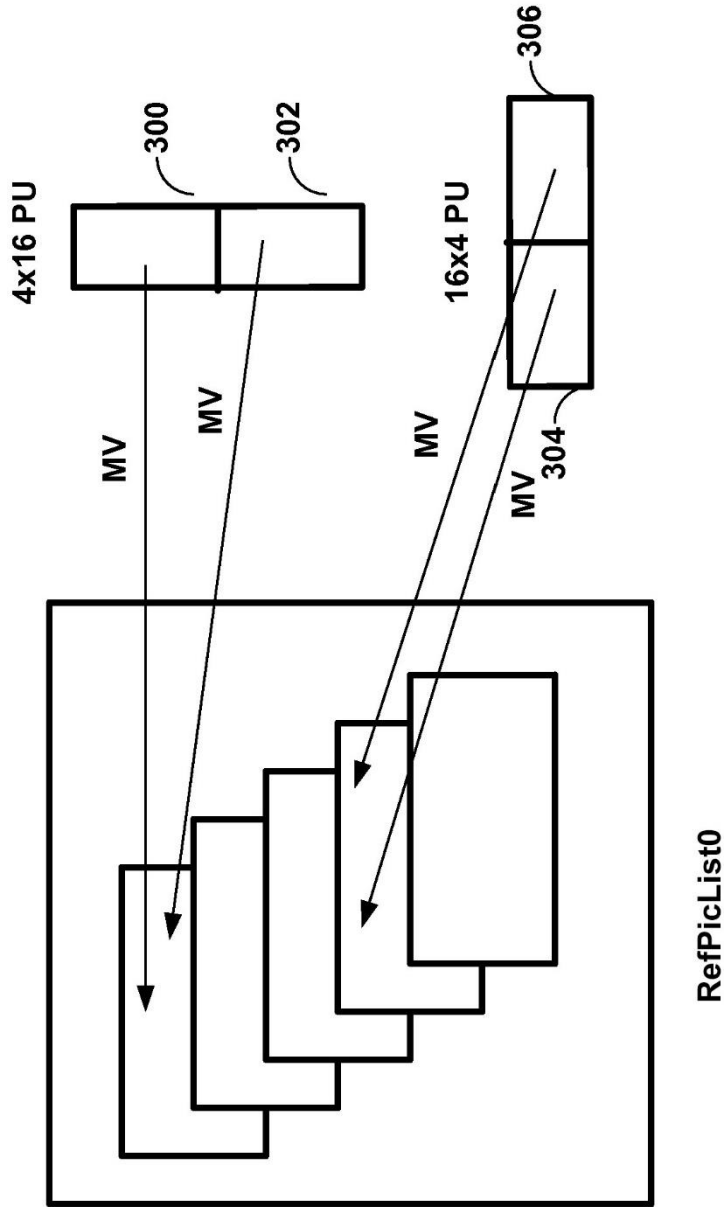


FIG. 18

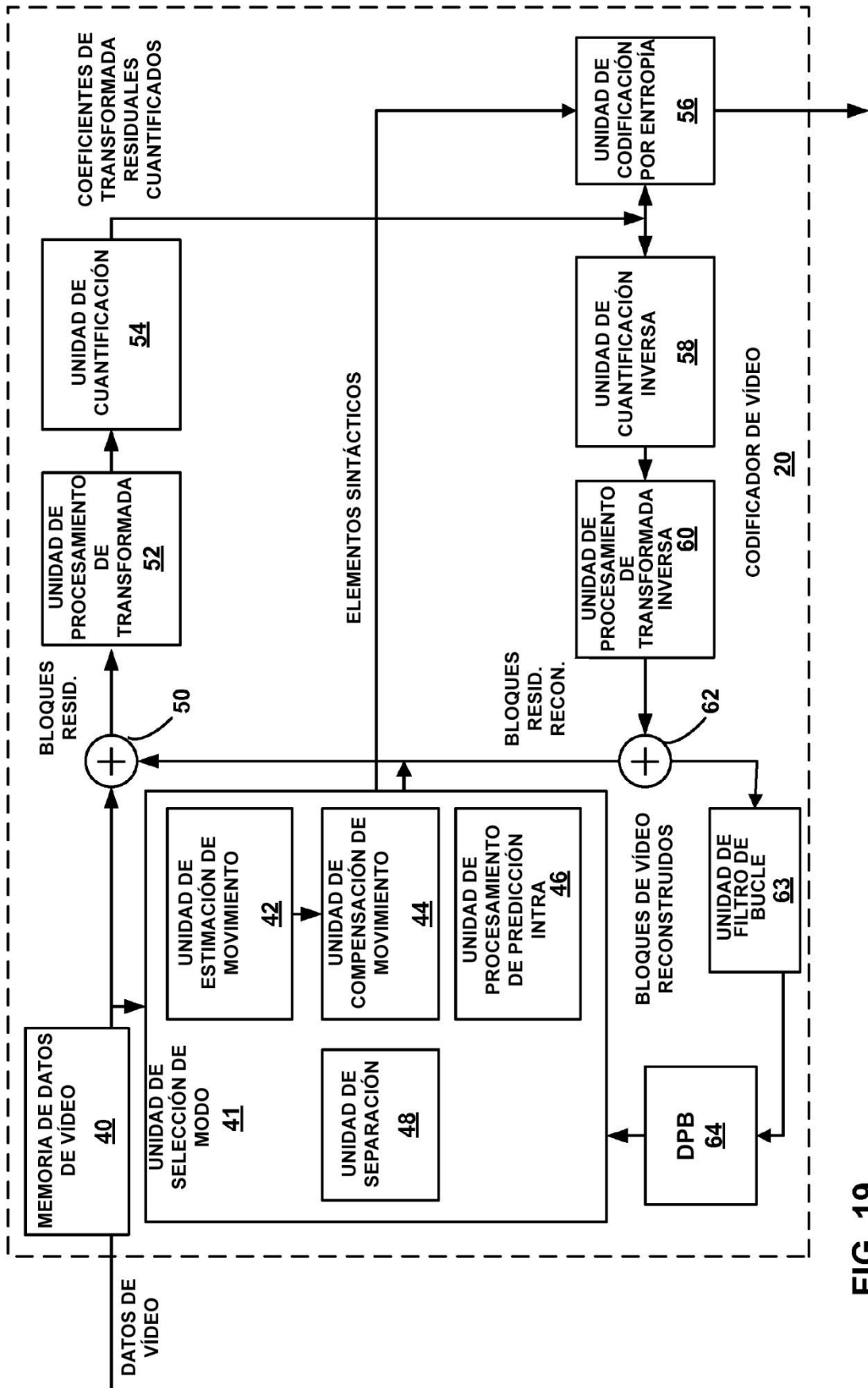


FIG. 19

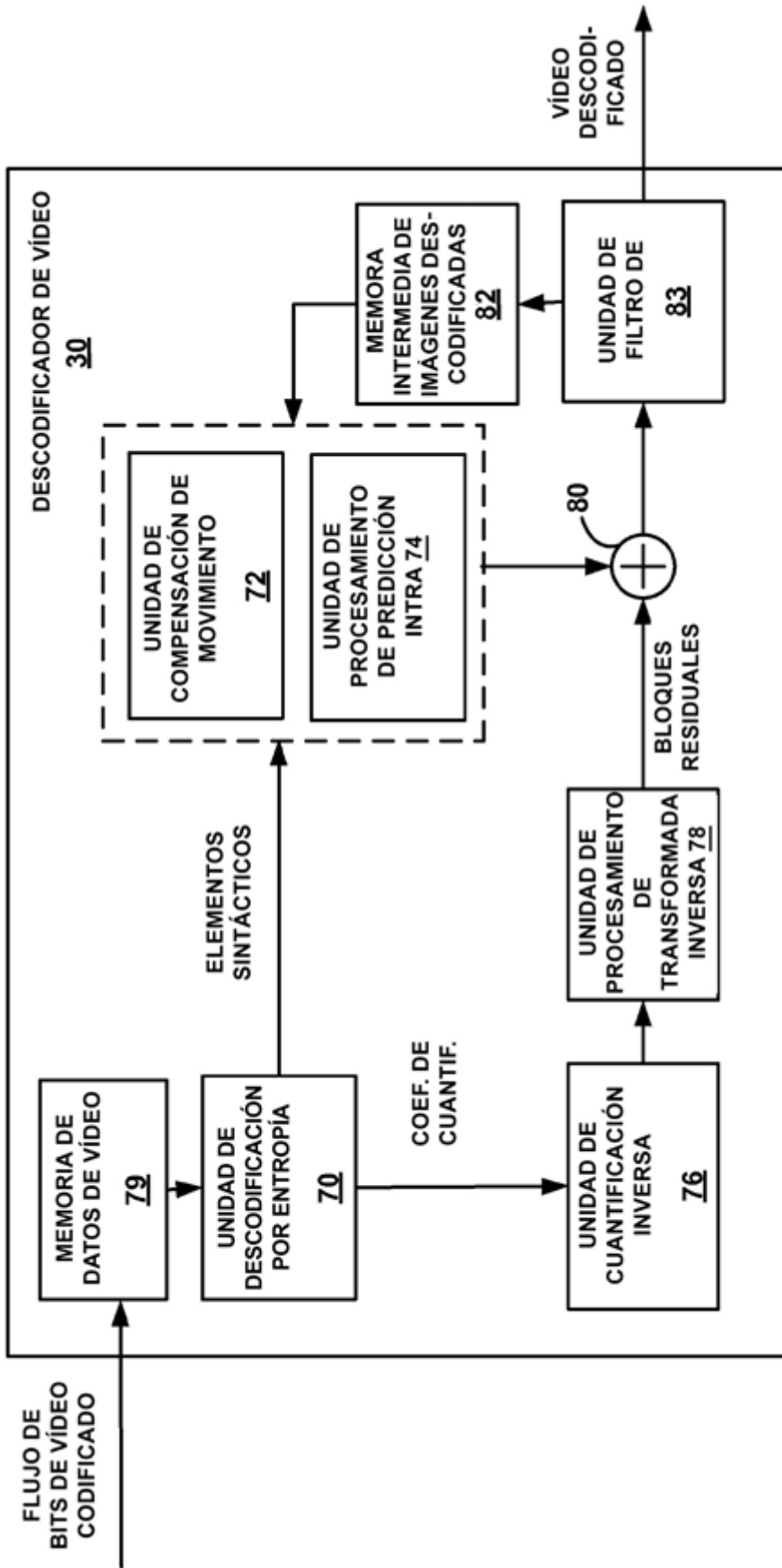


FIG. 20

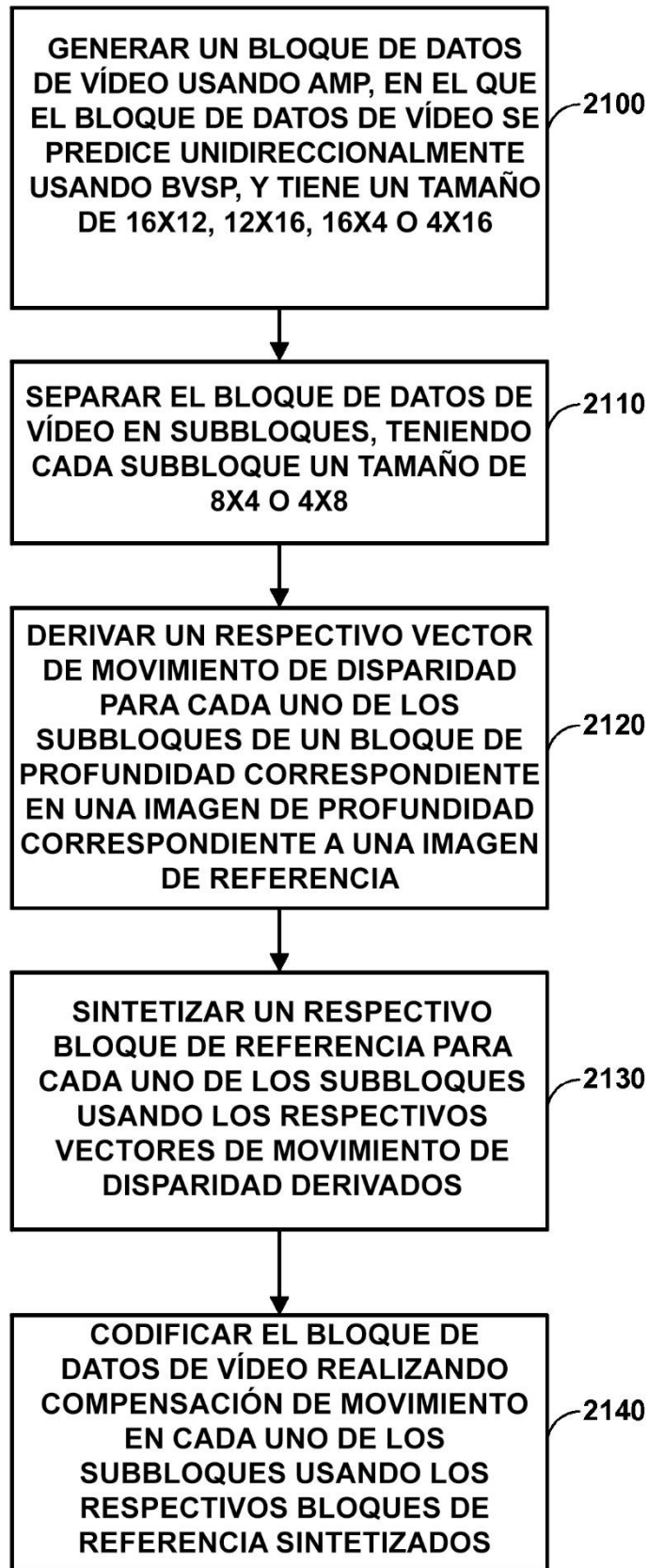


FIG. 21

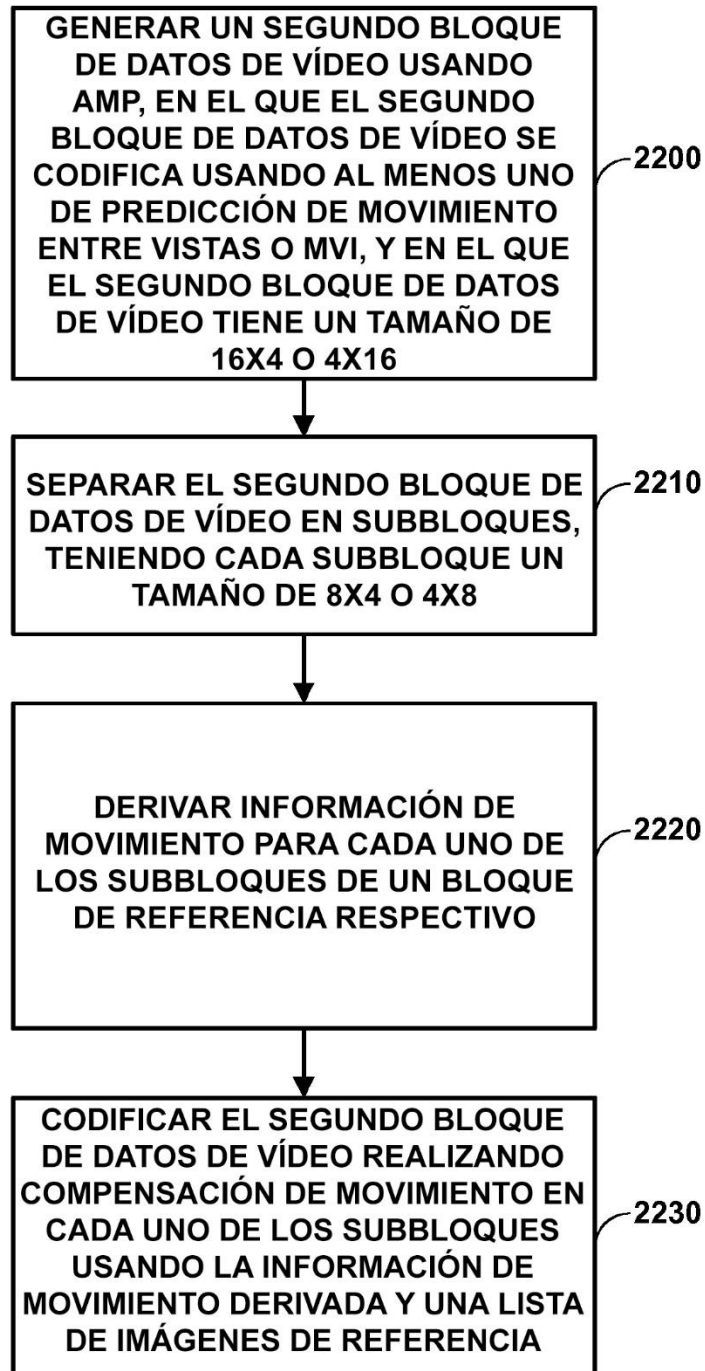


FIG. 22

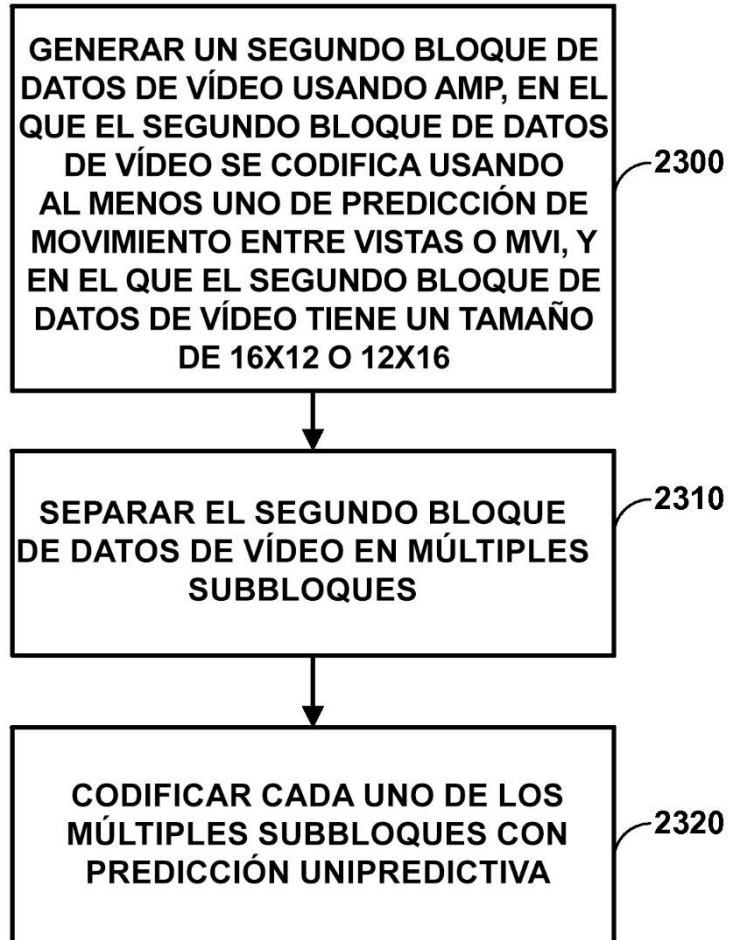


FIG. 23

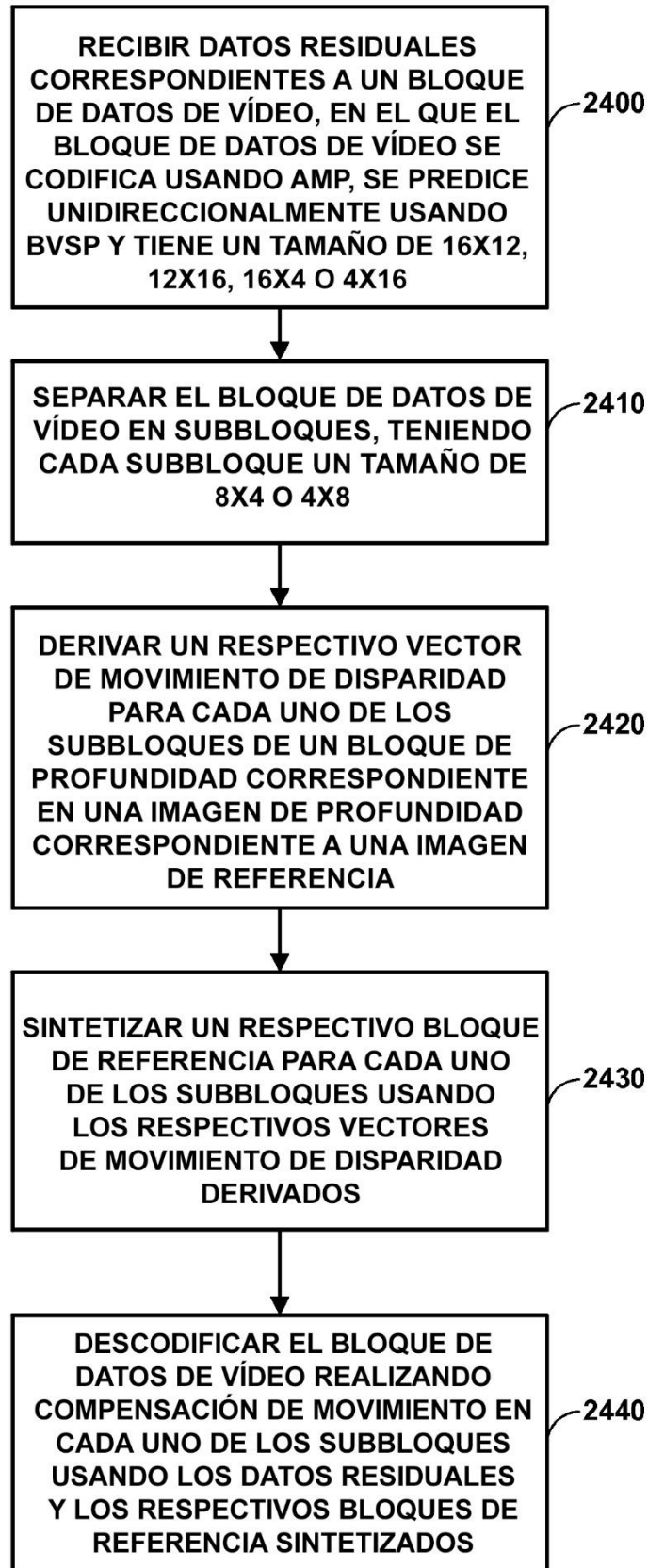


FIG. 24

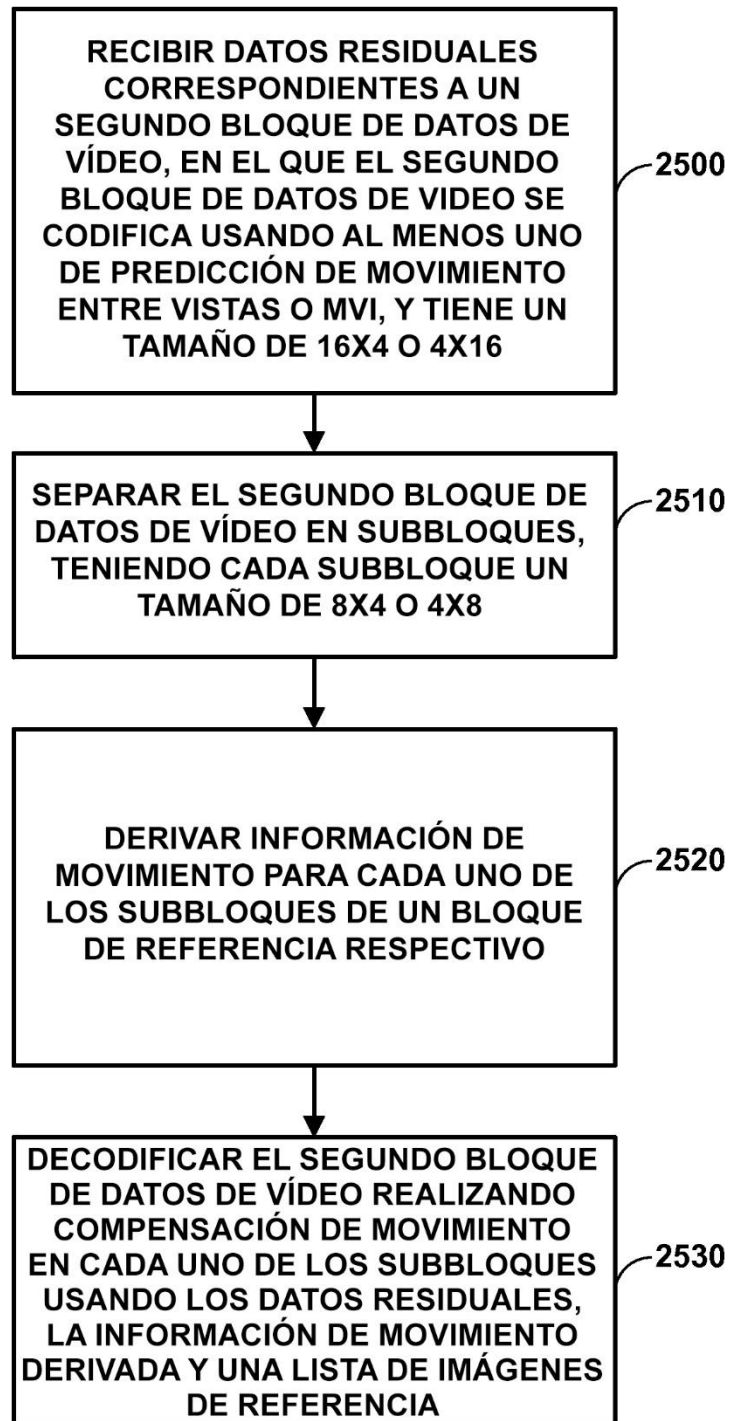


FIG. 25

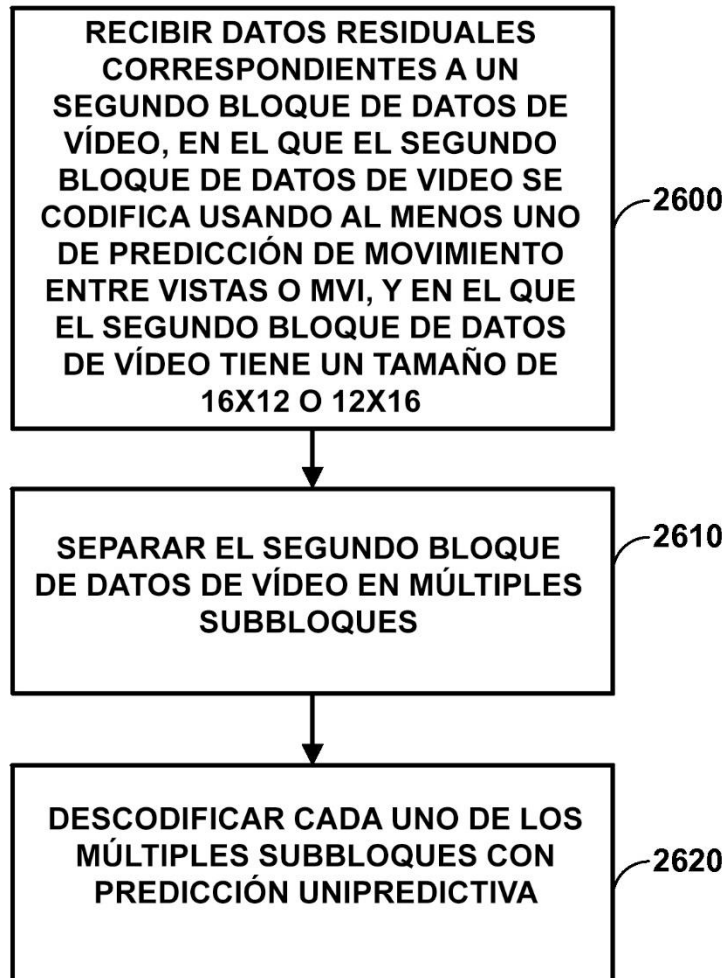


FIG. 26