

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 736 280**

51 Int. Cl.:

H04N 19/597 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **18.03.2014 PCT/US2014/031029**

87 Fecha y número de publicación internacional: **25.09.2014 WO14153340**

96 Fecha de presentación y número de la solicitud europea: **18.03.2014 E 14724873 (6)**

97 Fecha y número de publicación de la concesión europea: **17.04.2019 EP 2976885**

54 Título: **Simplificaciones en la obtención de vectores de disparidad y la predicción de vectores de movimiento en codificación de vídeo 3D**

30 Prioridad:

18.03.2013 US 201361803100 P
01.04.2013 US 201361807281 P
17.03.2014 US 201414216232

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
27.12.2019

73 Titular/es:

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

72 Inventor/es:

THIRUMALAI, VIJAYARAGHAVAN;
ZHANG, LI;
CHEN, YING y
KARCZEWICZ, MARTA

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 736 280 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

Simplificaciones en la obtención de vectores de disparidad y la predicción de vectores de movimiento en codificación de vídeo 3D

5 [0001] Esta aplicación reivindica el beneficio de la solicitud provisional de Estados Unidos n.º 61/803 100, presentada el 18 de marzo de 2013, y de la solicitud provisional de Estados Unidos n.º 61/807 281, presentada el 1 de abril de 2013.

10 CAMPO TÉCNICO

[0002] Esta divulgación se refiere a la codificación y a la decodificación de vídeo.

15 ANTECEDENTES

15 [0003] Las capacidades del vídeo digital pueden incorporarse a una amplia gama de dispositivos, incluidos televisores digitales, sistemas de difusión directa digital, sistemas de difusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de sobremesa, cámaras digitales, dispositivos de grabación digitales, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos celulares o de radio por satélite, dispositivos de videoconferencia y similares. Los dispositivos de vídeo digital implementan técnicas de compresión de vídeo, tales como las descritas en las normas definidas por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificación Avanzada de Vídeo (AVC), la norma de Codificación de Vídeo de Alta Eficacia (HEVC), actualmente en desarrollo, y las extensiones de dichas normas, para transmitir, recibir y almacenar información de vídeo digital de forma más eficaz.

25 [0004] Las extensiones de algunas de las normas mencionadas anteriormente, incluyendo la H.264/AVC, pueden proporcionar técnicas para la codificación de vídeo de visualización múltiple para producir vídeo estéreo o tridimensional ("3D"). En particular, se han propuesto técnicas para la codificación de visualización múltiple para su uso en la AVC, con la norma de codificación de vídeo ajustable a escala (SVC) (que es la ampliación ajustable a escala para la H.264/AVC), y la norma de codificación de vídeo de visualización múltiple (MVC) (que se ha convertido en la ampliación de visualización múltiple para la H.264/AVC).

35 [0005] Habitualmente, el vídeo estéreo se logra usando dos visualizaciones, por ejemplo, una visualización izquierda y una visualización derecha. Una imagen de la visualización izquierda se puede exhibir de forma esencialmente simultánea con una imagen de la visualización derecha para lograr un efecto de vídeo tridimensional. Por ejemplo, un usuario puede usar gafas pasivas polarizadas que filtran la visualización izquierda de la visualización derecha. De forma alternativa, las imágenes de las dos visualizaciones se pueden mostrar en rápida sucesión, y el usuario puede usar gafas activas con obturación rápida de los ojos izquierdo y derecho a la misma frecuencia, pero con un desplazamiento de fase de 90 grados.

40 [0006] DMYTRO RUSANOVSKYY ET AL: "CE1.a-related: Simplification of BVSP in 3DV-ATM [Relacionado con CE1.a: simplificación de BVSP en 3DV-ATM]", 3. REUNIÓN DE JCT-3V; 103. REUNIÓN DE MPEG; 17-1-2013 - 23-1-2013; GINEBRA; (EQUIPO DE COLABORACIÓN CONJUNTA EN DESARROLLO DE EXTENSIONES DE CODIFICACIÓN DE VÍDEO EN 3D DE ISO/IEC JTC1/SC29/WG11 E ITU-T SG.16); URL: HTTP://PHENIX.INT-EVRY.FR/JCT2/, n.º JCT3V-C0169, el 10 de enero de 2013 propone determinar, en el modo BVSP, los vectores de disparidad para bloques de tamaño 4x4 y 8x8.

SUMARIO

50 [0007] En general, esta divulgación describe técnicas para la codificación de vídeo 3D. En particular, esta divulgación está relacionada con técnicas para obtener vectores de disparidad.

55 [0008] En un ejemplo, un procedimiento de codificación de datos de vídeo 3D incluye la determinación de un primer bloque de una primera visualización de textura que se va a codificar utilizando un modo de síntesis de visualización basada en bloques; ubicar, en una visualización en profundidad, un primer bloque de profundidad que corresponde al primer bloque de la primera visualización de textura; determinar los valores de profundidad de dos o más posiciones de esquina del primer bloque de profundidad; basándose en los valores de profundidad, obtener un vector de disparidad para el primer bloque; utilizando el vector de disparidad, ubicar un primer bloque de una segunda visualización de textura; y realizar predicción inter del primer bloque de la primera visualización de textura utilizando el primer bloque de la segunda visualización de textura.

60 [0009] En otro ejemplo, un dispositivo para codificar datos de vídeo 3D incluye un codificador de vídeo configurado para determinar que un primer bloque de una primera visualización de textura debe codificarse utilizando un modo de síntesis de visualización basada en bloques; ubicar, en una visualización en profundidad, un primer bloque de profundidad que corresponde al primer bloque de la primera visualización de textura; determinar los valores de profundidad de dos o más posiciones de esquina del primer bloque de profundidad; basándose en los valores de

profundidad, obtener un vector de disparidad para el primer bloque; utilizando el vector de disparidad, ubicar un primer bloque de una segunda visualización de textura; y realizar predicción inter del primer bloque de la primera visualización de textura usando el primer bloque de la segunda visualización de textura.

5 **[0010]** En otro ejemplo, un dispositivo para codificar datos de vídeo 3D incluye medios para determinar un primer bloque de una primera visualización de textura que se va a codificar usando un modo de síntesis de visualización basada en bloques; medios para ubicar, en una visualización en profundidad, un primer bloque de profundidad que corresponde al primer bloque de la primera visualización de textura; medios para determinar valores de profundidad de dos o más posiciones de esquina del primer bloque de profundidad; basándose en los valores de profundidad,
10 medios para obtener un vector de disparidad para el primer bloque; utilizando el vector de disparidad, medios para ubicar un primer bloque de una segunda visualización de textura; y, medios para predecir el primer bloque de la primera visualización de textura usando el primer bloque de la segunda visualización de textura.

15 **[0011]** En otro ejemplo, un medio de almacenamiento legible por ordenador almacena instrucciones que cuando son ejecutadas por uno o más procesadores hacen que uno o más procesadores determinen que un primer bloque de una primera visualización de textura debe codificarse utilizando un modo de síntesis de visualización basada en bloques; ubicar, en una visualización en profundidad, un primer bloque de profundidad que corresponde al primer bloque de la primera visualización de textura; determinar los valores de profundidad de dos o más posiciones de esquina del primer bloque de profundidad; basándose en los valores de profundidad, obtener un vector de disparidad para el primer
20 bloque; utilizando el vector de disparidad, ubicar un primer bloque de una segunda visualización de textura; y realizar predicción inter del primer bloque de la primera visualización de textura utilizando el primer bloque de la segunda visualización de textura.

25 **[0012]** Los detalles de uno o más ejemplos se exponen en los dibujos adjuntos y en la siguiente descripción. Otras características, objetos y ventajas resultarán evidentes a partir de la descripción y de los dibujos, y a partir de las reivindicaciones.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

30 **[0013]**

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

35 La FIG. 2 es un diagrama conceptual que ilustra un orden a modo de ejemplo de decodificación de visualización múltiple.

La FIG. 3 es un diagrama conceptual que ilustra una estructura a modo de ejemplo de predicción para la codificación de visualización múltiple.

40 La FIG. 4 es una visualización conceptual de la predicción de la síntesis de visualización basada en bloques basándose en la distorsión inversa.

45 La FIG. 5 es un diagrama conceptual que muestra bloques contiguos espaciales usados para la obtención de vectores de disparidad de bloque contiguo.

La FIG. 6 es un diagrama conceptual que ilustra técnicas relacionadas con la predicción de síntesis de visualización hacia atrás (BVSP) utilizando bloques contiguos.

50 La FIG. 7 muestra un ejemplo de un bloque de profundidad que puede usarse para obtener un vector de disparidad de acuerdo con las técnicas de esta divulgación.

La FIG. 8 es un diagrama de bloques que ilustra un codificador de vídeo a modo de ejemplo que puede implementar las técnicas descritas en esta divulgación.

55 La FIG. 9 es un diagrama de bloques que ilustra un decodificador de vídeo a modo de ejemplo que puede implementar las técnicas descritas en esta divulgación.

60 La FIG. 10 es un diagrama de flujo que ilustra una operación de ejemplo de obtención de vector de disparidad y codificación entre predicciones de acuerdo con las técnicas descritas en esta divulgación.

La FIG. 11 es un diagrama de flujo que ilustra otra operación de ejemplo de obtención de vector de disparidad y codificación entre predicciones de acuerdo con las técnicas descritas en esta divulgación.

65

DESCRIPCIÓN DETALLADA

[0014] Esta divulgación describe técnicas relacionadas con la obtención de vector de disparidad, y más particularmente, esta divulgación describe técnicas en las que un codificador de vídeo (por ejemplo, el codificador de vídeo o el descodificador de vídeo) obtiene un vector de disparidad para un bloque actual dentro de una imagen actual de una visualización actual para uso en codificación de vídeo tridimensional (3D). El codificador de vídeo puede usar el vector de disparidad para ubicar un bloque correspondiente en una visualización diferente. De esta manera, el vector de disparidad puede representar la disparidad entre dos bloques de vídeo similares en dos visualizaciones diferentes. Como se explicará con mayor detalle a continuación, el codificador de vídeo puede usar el vector de disparidad como un vector de movimiento de disparidad, y se puede predecir un bloque actual basándose en el bloque que se encuentra ubicado en el vector de movimiento de disparidad. El codificador de vídeo también puede usar el vector de disparidad para otros propósitos. Como ejemplo, el codificador de vídeo puede usar el vector de disparidad para ubicar un bloque correspondiente en otra visualización y luego usar la información de movimiento del bloque ubicado para determinar la información de movimiento del bloque actual. Además, puede haber otros usos para el vector de disparidad. En esta divulgación, el término "actual" se utiliza en general para referirse a una visualización, imagen o bloque que se está codificando actualmente. De este modo, un bloque actual en general representa un bloque de datos de vídeo codificados, en oposición a un bloque de vídeo ya codificado o en oposición a un bloque de vídeo no codificado todavía.

[0015] Un vector de disparidad para un bloque actual de una imagen actual es un vector que apunta a un bloque correspondiente en una imagen correspondiente que está en una visualización diferente a la imagen actual. Por tanto, utilizando un vector de disparidad, un codificador de vídeo puede ubicar, en una imagen correspondiente, el bloque que corresponde a un bloque actual de una imagen actual. En este caso, la imagen correspondiente es una imagen que es de la misma instancia temporal que la imagen actual pero que está en una visualización diferente. El bloque correspondiente en la imagen correspondiente y el bloque actual en la imagen actual pueden incluir contenido de vídeo similar; sin embargo, hay al menos una disparidad horizontal entre la ubicación del bloque actual en la imagen actual y la ubicación del bloque correspondiente en la imagen correspondiente. El vector de disparidad del bloque actual proporciona una medida de esta disparidad horizontal entre el bloque en la imagen correspondiente y el bloque actual en la imagen actual. En algunos casos, también puede haber disparidad vertical entre la ubicación del bloque dentro de la imagen correspondiente y la ubicación del bloque actual dentro de la imagen actual; sin embargo, en muchos casos, la disparidad vertical será cero. El vector de disparidad del bloque actual también puede proporcionar una medida de esta disparidad vertical entre el bloque en la imagen correspondiente y el bloque actual en la imagen actual. Un vector de disparidad contiene dos componentes (un componente x y un componente y), aunque en muchos casos el componente vertical será igual a cero. El tiempo en que se muestra la imagen actual de la visualización actual y la imagen correspondiente de la visualización diferente puede ser el mismo, es decir, la imagen actual y las imágenes correspondientes son imágenes de la misma instancia temporal.

[0016] En codificación de vídeo 2D una trama está representada solamente por un componente de visualización, a veces denominado componente de visualización de textura o, simplemente, textura. En algunos tipos de codificación de vídeo 3D, existen dos componentes de visualización: un componente de visualización de textura y un componente de visualización de profundidad o, simplemente, textura y profundidad. Por ejemplo, cada visualización puede incluir una visualización de textura y una visualización de profundidad, donde la visualización incluye una pluralidad de componentes de visualización; por ejemplo, la visualización de textura incluye una pluralidad de componentes de visualización de textura y la visualización de profundidad incluye una pluralidad de componentes de visualización de profundidad. Cada componente de visualización de textura está asociado a un componente de visualización de profundidad para formar un componente de visualización de una visualización. El componente de visualización de profundidad representa la profundidad relativa de los objetos en el componente de visualización de textura. El componente de visualización de profundidad y el componente de visualización de textura pueden ser descodificables por separado.

[0017] Esta divulgación describe técnicas para obtener vectores de disparidad. Una técnica de este tipo para obtener un vector de disparidad se puede utilizar junto con un modo de predicción de síntesis hacia atrás (BVSP). El codificador de vídeo puede determinar que un bloque de una primera visualización de textura debe codificarse utilizando un modo BVSP. El codificador de vídeo puede ubicar, en una visualización de profundidad, un bloque de profundidad que corresponde al bloque de la primera visualización de textura y determinar los valores de profundidad para dos o más posiciones de esquina del bloque de profundidad. Basándose en los valores de profundidad, el codificador de vídeo puede obtener un vector de disparidad para el bloque, y utilizando el vector de disparidad, ubicar un bloque de una segunda visualización de textura. El codificador de vídeo puede entonces predecir el bloque de la primera visualización de textura usando el bloque de la segunda visualización de textura. A este respecto, y como se explicará con más detalle a continuación, las técnicas de esta divulgación pueden simplificar el proceso de obtención de vectores de disparidad al determinar un vector de disparidad, para uso en el modo BVSP, usando solo las muestras de esquina del bloque de profundidad correspondiente.

[0018] En otra técnica de ejemplo, para un bloque de una primera visualización, un codificador de vídeo puede ubicar un bloque de profundidad en una visualización de profundidad que corresponda al bloque de la primera visualización de textura, y basándose en al menos un valor de profundidad del bloque de profundidad, obtener un vector de disparidad para el bloque de la primera visualización de textura. El codificador de vídeo puede entonces codificar un primer sub-bloque del bloque basándose en el vector de disparidad obtenido y codificar un segundo sub-

bloque del macrobloque basándose en el mismo vector de disparidad obtenido. A este respecto y como se explicará con más detalle a continuación, las técnicas de esta divulgación pueden simplificar el proceso de obtención de vector de disparidad al obtener un vector de disparidad para un bloque y usar esa disparidad para dos o más sub-bloques del bloque. Dependiendo del modo de codificación especificado para un sub-bloque particular, el codificador de vídeo puede usar el vector de disparidad obtenido como un vector de movimiento de disparidad o puede usar el vector de disparidad para identificar un bloque correspondiente en una visualización diferente, y a partir de ese bloque correspondiente, determinar Información de movimiento para predecir el sub-bloque.

[0019] La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y decodificación de vídeo de ejemplo 10 que puede configurarse para realizar las técnicas para obtención de vector de disparidad descritas en esta divulgación. Como se muestra en la FIG. 1, el sistema 10 incluye un dispositivo de origen 12 que genera datos de vídeo codificados que un dispositivo de destino 14 va a decodificar en un momento posterior. El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender cualquiera entre una amplia gama de dispositivos, incluyendo ordenadores de sobremesa, ordenadores plegables (es decir, portátiles), ordenadores de tipo tableta, descodificadores, equipos telefónicos tales como los denominados teléfonos "inteligentes", los denominados paneles "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de transmisión continua de vídeo o similares. En algunos casos, el dispositivo de origen 12 y el dispositivo de destino 14 pueden estar equipados para la comunicación inalámbrica.

[0020] El sistema 10 puede funcionar de acuerdo con diferentes normas de codificación de vídeo, una norma patentada o cualquier otra forma de codificación de visualización múltiple. A continuación se describen unos pocos ejemplos de normas de codificación de vídeo, y no se deben considerar limitantes. Entre las normas de codificación de vídeo se incluyen ITU-T H.261, MPEG-1 Visual de la ISO/IEC, ITU-T H.262 o MPEG-2 Visual de la ISO/IEC, ITU-T H.263, MPEG-4 Visual de la ISO/IEC e ITU-T H.264 (también conocida como AVC del MPEG-4 de la ISO/IEC), incluyendo sus ampliaciones de codificación de vídeo escalable (SVC) y de codificación de vídeo de múltiples visualizaciones (MVC). El más reciente borrador conjunto de la MVC se describe en "Advanced vídeo coding for generic audiovisual services [Codificación de vídeo avanzada para servicios audiovisuales genéricos]", Recomendación H.264 de la ITU-T, marzo de 2010. Otro borrador conjunto de la MVC se describe en "Advanced vídeo coding for generic audiovisual services [Codificación de vídeo avanzada para servicios audiovisuales genéricos]", Recomendación ITU-T H.264, junio de 2011. Algunas normas adicionales de codificación de vídeo incluyen la MVC+D y la 3D-AVC, que se basan en la AVC. Además, una nueva norma de codificación de vídeo, concretamente, la Codificación de Vídeo de Alta Eficacia (HEVC), ha sido desarrollada por el Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) del Grupo de Expertos en Codificación de Vídeo (VCEG) de la ITU-T y el Grupo de Expertos en Imágenes en Movimiento (MPEG) de ISO/IEC.

[0021] Con fines de ilustración solamente, las técnicas descritas en esta divulgación se describen con ejemplos de acuerdo con la norma H.264, tales como la 3D-AVC. Sin embargo, las técnicas descritas en esta divulgación no se deberían considerar limitadas a estas normas a modo de ejemplo, y se pueden ampliar a otras normas de codificación de vídeo para la codificación de visualización múltiple o la codificación de vídeo 3D (por ejemplo, la 3D-HEVC), o a técnicas relacionadas con la codificación de visualización múltiple o la codificación de vídeo 3D, que no se basan necesariamente en una norma de codificación de vídeo particular. Por ejemplo, las técnicas descritas en esta divulgación se implementan mediante codificadores/descodificadores de vídeo (códecs) para la codificación de visualización múltiple, donde la codificación de visualización múltiple incluye la codificación de dos o más visualizaciones.

[0022] El dispositivo de destino 14 puede recibir los datos de vídeo codificados que se van a decodificar a través de un enlace 16. El enlace 16 puede comprender cualquier tipo de medio o dispositivo capaz de desplazar los datos de vídeo codificados desde el dispositivo de origen 12 hasta el dispositivo de destino 14. En un ejemplo, el enlace 16 puede comprender un medio de comunicación para permitir que el dispositivo de origen 12 transmita datos de vídeo codificados directamente a un 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 cableada, tal como un espectro de radiofrecuencia (RF) o una o más líneas físicas de transmisión. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área amplia o una red global tal como Internet. El medio de comunicación puede incluir encaminadores, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo de origen 12 hasta el dispositivo de destino 14.

[0023] De forma alternativa, los datos codificados pueden emitirse desde la interfaz de salida 22 hasta un dispositivo de almacenamiento 34. De forma similar, una interfaz de entrada puede acceder a los datos codificados procedentes del dispositivo de almacenamiento 34. El dispositivo de almacenamiento 34 puede incluir cualquiera de una variedad de medios de almacenamiento de datos de acceso distribuido o local, tales como una unidad de disco duro, unos discos Blu-ray, unos DVD, unos CD-ROM, una memoria flash, una memoria volátil o no volátil o cualquier otro medio de almacenamiento digital adecuado para almacenar datos de vídeo codificados. En otro ejemplo, el dispositivo de almacenamiento 34 puede corresponder a un servidor de archivos o a otro dispositivo de almacenamiento intermedio que puede contener el vídeo codificado generado por el dispositivo de origen 12. El dispositivo de destino 14 puede

acceder a datos de vídeo almacenados en el dispositivo de almacenamiento 34 a través de transmisión en tiempo real o descarga. El servidor de ficheros puede ser cualquier tipo de servidor capaz de almacenar datos de vídeo codificados y transmitir esos datos de vídeo codificados al dispositivo de destino 14. Los servidores a modo de ejemplo de ficheros incluyen un servidor de la Red (por ejemplo, para una sede de la Red), un servidor del FTP, dispositivos de almacenamiento conectados en red (NAS) o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de cualquier conexión de datos estándar, incluida una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión de Wi-Fi), una conexión cableada (por ejemplo, DSL, módem de cable, etc.) o una combinación de 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 34 puede ser una transmisión en tiempo real, una transmisión de descarga o una combinación de ambas.

[0024] Las técnicas de esta divulgación para obtención de vector de disparidad no se limitan necesariamente a aplicaciones o configuraciones inalámbricas. Las técnicas se pueden aplicar a la codificación de vídeo como apoyo a cualquiera de una variedad de aplicaciones multimedia, tales como radiodifusiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo en tiempo real, por ejemplo, a través de Internet, codificación de vídeo digital para su almacenamiento en un medio de almacenamiento de datos, descodificación de vídeo digital almacenado en un medio de almacenamiento de datos, u otras aplicaciones. En algunos ejemplos, el sistema 10 puede estar configurado para prestar soporte a una transmisión de vídeo unidireccional o bidireccional, a fin de prestar soporte a aplicaciones tales como la transmisión de vídeo en tiempo real, la reproducción de vídeo, la difusión de vídeo y/o la videotelefonía.

[0025] En el ejemplo de la FIG. 1, el dispositivo de origen 12 incluye una fuente de vídeo 18, un codificador de vídeo 20 y una interfaz de salida 22. En algunos casos, la interfaz de salida 22 puede incluir un modulador/desmodulador (módem) y/o un transmisor. En el dispositivo de origen 12, la fuente de vídeo 18 puede incluir una fuente tal como un dispositivo de captación de vídeo, por ejemplo, una videocámara, un archivo de vídeo que contiene vídeo captado previamente, una interfaz de señal de vídeo en tiempo real para recibir vídeo desde un proveedor de contenido de vídeo y/o un sistema de gráficos por ordenador para generar datos de gráficos por ordenador como el vídeo de origen, o una combinación de dichas fuentes. En un ejemplo, si el origen de vídeo 18 es una videocámara, el dispositivo de origen 12 y el dispositivo de destino 14 pueden formar los denominados teléfonos con cámara o videoteléfonos. Sin embargo, las técnicas descritas en esta divulgación pueden ser aplicables a la codificación de vídeo en general, y se pueden aplicar a aplicaciones inalámbricas y/o alámbricas.

[0026] El codificador de vídeo 20 puede codificar el vídeo capturado, capturado previamente o generado por ordenador. Los datos de vídeo codificados se pueden transmitir directamente al dispositivo de destino 14 por medio de la interfaz de salida 22 del dispositivo de origen 12. Los datos de vídeo codificados se pueden almacenar de forma adicional (o alternativa) en el dispositivo de almacenamiento 34 para un posterior acceso por el dispositivo de destino 14 u otros dispositivos, para su descodificación y/o reproducción.

[0027] El dispositivo de destino 14 incluye una interfaz de entrada 28, un descodificador de vídeo 30 y un dispositivo de visualización 32. En algunos casos, la interfaz de entrada 28 puede incluir un receptor y/o un módem. La interfaz de entrada 28 del dispositivo de destino 14 recibe los datos de vídeo codificados a través del enlace 16. Los datos de vídeo codificados transmitidos a través del enlace 16, o proporcionados en el dispositivo de almacenamiento 34, pueden incluir una diversidad de elementos sintácticos generados por el codificador de vídeo 20 para su uso por un descodificador de vídeo, tal como el descodificador de vídeo 30, en la descodificación de los datos de vídeo. Dichos elementos sintácticos se pueden incluir con los datos de vídeo codificados transmitidos en un medio de comunicación, almacenados en un medio de almacenamiento o almacenados en un servidor de ficheros.

[0028] El dispositivo de visualización 32 puede estar integrado en, o ser externo a, el dispositivo de destino 14. En algunos ejemplos, el dispositivo de destino 14 puede incluir un dispositivo de visualización integrado y también estar configurado para interconectarse con un dispositivo de visualización externo. En otros ejemplos, el dispositivo de destino 14 puede ser un dispositivo de visualización. En general, el dispositivo de visualización 32 visualiza los datos de vídeo descodificados ante un usuario, y puede comprender cualquiera de una variedad de dispositivos de visualización, tales como una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de visualización.

[0029] Aunque no se muestra en la FIG. 1, en algunos aspectos, tanto el codificador de vídeo 20 como el descodificador de vídeo 30 pueden estar integrados con un codificador y descodificador de audio, y pueden incluir unidades MUX-DEMUX adecuadas, u otro tipo de hardware y software, para ocuparse de la codificación tanto de audio como de vídeo en un flujo de datos común o en flujos de datos separados. Si procede, en algunos ejemplos, las unidades MUX-DEMUX pueden ajustarse al protocolo de multiplexador ITU H.223 o a otros protocolos, tales como el protocolo de datagramas de usuario (UDP).

[0030] Tanto el codificador de vídeo 20 como el descodificador de vídeo 30 pueden implementarse como cualquiera entre varios circuitos codificadores adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), formaciones de puertas programables in situ

(FPGA), lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. Por ejemplo, las técnicas descritas en esta divulgación se pueden describir desde la perspectiva de un aparato o un dispositivo. Como ejemplo, el aparato o dispositivo puede incluir el descodificador de vídeo 30 (por ejemplo, el dispositivo de destino 14 como parte de un dispositivo de comunicación inalámbrica), y el descodificador de vídeo 30 puede incluir uno o más procesadores configurados para implementar las técnicas descritas en esta divulgación (por ejemplo, descodificar datos de vídeo de acuerdo con las técnicas descritas en esta divulgación). Como otro ejemplo, el aparato o dispositivo puede incluir un microprocesador o un circuito integrado (IC) que incluye el descodificador de vídeo 30, y el microprocesador o IC puede ser parte del dispositivo de destino 14 u otro tipo de dispositivo. Lo mismo se puede aplicar al codificador de vídeo 20 (es decir, un aparato o dispositivo como el dispositivo de origen 12 y/o un micro-controlador o IC incluye el codificador de vídeo 20, donde el codificador de vídeo 20 se configura para codificar datos de vídeo de acuerdo con las técnicas descritas en esta divulgación).

[0031] Cuando las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio adecuado no transitorio legible por ordenador, y ejecutar las instrucciones en hardware usando uno o más procesadores para realizar las técnicas de esta divulgación. Cada uno entre el codificador de vídeo 20 y el descodificador de vídeo 30 se puede incluir en uno o más codificadores o descodificadores, cualquiera de los cuales se puede integrar como parte de un codificador/descodificador (CÓDEC) combinado en un dispositivo respectivo.

[0032] Una secuencia de vídeo típicamente incluye una serie de imágenes de vídeo desde una visualización. Un grupo de imágenes (GOP) comprende en general una serie de una o más imágenes de vídeo. Un GOP puede incluir datos sintácticos en una cabecera del GOP, una cabecera de una o más imágenes del GOP o en otras ubicaciones, que describen una serie de imágenes incluidas en el GOP. Cada imagen puede incluir datos sintácticos de imagen que describe un modo de codificación para la imagen respectiva. Un codificador de vídeo 20 opera habitualmente sobre bloques de vídeo dentro de imágenes de vídeo individuales a fin de codificar los datos de vídeo. Un bloque de vídeo puede corresponder a un macrobloque, una partición de un macrobloque y, posiblemente, un sub-bloque de una partición, como se define en la norma H.264. Los bloques de vídeo pueden tener tamaños fijos o variables y pueden diferir en tamaño de acuerdo con una norma de codificación especificada. Cada imagen de vídeo puede incluir una pluralidad de fragmentos. Cada fragmento puede incluir una pluralidad de bloques.

[0033] Como ejemplo, la norma ITU-T H.264 presta soporte a la predicción intra en diversos tamaños de bloque, tales como 16 por 16, 8 por 8 o 4 por 4 para componentes de luma, y 8x8 para componentes de croma, así como la predicción inter en diversos tamaños de bloque, tales como 16x16, 16x8, 8x16, 8x8, 8x4, 4x8 y 4x4 para componentes de luma y tamaños correspondientes ajustados a escala para componentes de croma. En esta divulgación, "NxN" y "N por N" se pueden usar de manera intercambiable para referirse a las dimensiones de píxel del bloque en lo que respecta a las dimensiones vertical y horizontal, (por ejemplo, 16x16 píxeles o 16 por 16 píxeles). En general, un bloque 16x16 tendrá 16 píxeles en una dirección vertical ($y = 16$) y 16 píxeles en una dirección horizontal ($x = 16$). Asimismo, un bloque de tamaño NxN tiene, en general, N píxeles en una dirección vertical y N píxeles en una dirección horizontal, donde N representa un valor entero no negativo. Los píxeles de un bloque se pueden disponer en filas y columnas. Además, no es necesario que los bloques tengan necesariamente el mismo número de píxeles en la dirección horizontal que en la dirección vertical. Por ejemplo, los bloques pueden comprender NxM píxeles, donde M no es necesariamente igual a N.

[0034] Cuando el bloque se codifica en modo intra (por ejemplo, predicción intra), el bloque puede incluir datos que describen un modo de predicción intra para el bloque. Como otro ejemplo, cuando el bloque se codifica en modo inter (por ejemplo, predicción inter), el bloque puede incluir información que define un vector de movimiento para el bloque. Este vector de movimiento se refiere a una imagen de referencia en la misma visualización (por ejemplo, un vector de movimiento temporal), o se refiere a una imagen de referencia en otra visualización (por ejemplo, un vector de movimiento de disparidad). Los datos que definen el vector de movimiento para un bloque describen, por ejemplo, un componente horizontal del vector de movimiento, un componente vertical del vector de movimiento, una resolución para el vector de movimiento (por ejemplo, una precisión de un cuarto de píxel o una precisión de un octavo de píxel). Además, cuando tiene predicción inter, el bloque puede incluir información del índice de referencia, tal como una imagen de referencia a la que apunta el vector de movimiento, y/o una lista de imágenes de referencia (por ejemplo, RefPicList0 o RefPicList1) para el vector de movimiento.

[0035] En la norma H.264, tras la codificación de predicción intra o predicción inter, el codificador de vídeo 20 calcula los datos residuales para los macrobloques. Los datos residuales pueden corresponder a diferencias de píxeles entre píxeles de la imagen no codificada y valores de predicción para el macrobloque en la norma H.264.

[0036] Tras cualquier transformada para producir coeficientes de transformada, el codificador de vídeo 20 realiza la cuantificación de los coeficientes de transformada, en algunos ejemplos. La cuantificación se refiere, en general, a un proceso en el que unos coeficientes de transformada se cuantifican para reducir posiblemente la cantidad de datos usados para representar los coeficientes, proporcionando compresión adicional. El proceso de cuantificación reduce la profundidad de bits asociada a algunos o a la totalidad de los coeficientes. Por ejemplo, un valor de n bits se redondea hacia abajo a un valor de m bits durante la cuantificación, donde n es mayor que m.

[0037] En algunos ejemplos, el codificador de vídeo 20 utiliza un orden de recorrido predefinido para explorar los coeficientes de transformada cuantificados, para producir un vector en serie que se pueda someter a codificación por entropía. En otros ejemplos, el codificador de vídeo 20 realiza un recorrido adaptativo. Después de recorrer los coeficientes de transformada cuantificados para formar un vector unidimensional, en algunos ejemplos, el codificador de vídeo 20 codifica por entropía el vector unidimensional de acuerdo con la codificación de longitud variable adaptativa al contexto (CAVLC), la codificación aritmética binaria adaptativa al contexto (CABAC), la codificación aritmética binaria adaptativa al contexto basada en la sintaxis (SBAC), la codificación por entropía por división de intervalos de probabilidad (PIPE) u otra metodología de codificación por entropía, como unos pocos ejemplos. El codificador de vídeo 20 también codifica por entropía elementos sintácticos asociados a los datos de vídeo codificados, para su uso por el descodificador de vídeo 30 en la descodificación de los datos de vídeo.

[0038] Para realizar la codificación CABAC, el codificador de vídeo 20 puede asignar un contexto, dentro de un modelo de contexto, a un símbolo a transmitir. El contexto se puede referir, por ejemplo, a si los valores contiguos del símbolo son distintos de cero o no. Para realizar la CAVLC, el codificador de vídeo 20 puede seleccionar un código de longitud variable para un símbolo a transmitir. Las palabras de código en la VLC se pueden construir de modo que los códigos relativamente más cortos correspondan a símbolos más probables, mientras que los códigos más largos correspondan a símbolos menos probables. De esta forma, el uso de la VLC puede lograr un ahorro en bits con respecto, por ejemplo, al uso de palabras de código de igual longitud para cada símbolo que se va a transmitir. La determinación de la probabilidad se puede basar en un contexto asignado al símbolo.

[0039] El descodificador de vídeo 30 implementa la inversa de las técnicas del codificador de vídeo 20. Por ejemplo, el descodificador de vídeo 30 descodifica el flujo de bits de vídeo codificado y determina los bloques residuales mediante cuantificación inversa y transformada inversa. El descodificador de vídeo 30 suma los bloques residuales con bloques de imágenes descodificadas previamente para determinar los valores de píxeles para los bloques dentro de la imagen.

[0040] Ciertas técnicas descritas en esta divulgación pueden ser realizadas tanto por el codificador de vídeo 20 como por el descodificador de vídeo 30. Como ejemplo, el codificador de vídeo 20 puede obtener un vector de disparidad como parte de la determinación de cómo codificar un bloque de datos de vídeo. El codificador de vídeo 20 también puede obtener el vector de disparidad como parte de un bucle de descodificación utilizado para generar imágenes de referencia. El descodificador de vídeo 30 puede realizar la misma técnica de obtención de vector de disparidad realizada por el codificador de vídeo 20 como parte de la descodificación del bloque de vídeo. Esta divulgación puede referirse a veces a técnicas realizadas por el descodificador de vídeo 30, pero a menos que se especifique lo contrario, debe suponerse que las técnicas descritas con referencia al descodificador de vídeo 30 también pueden ser realizadas por el codificador de vídeo 20.

[0041] Como se ha descrito anteriormente, las técnicas descritas en esta divulgación están dirigidas a la codificación de vídeo 3D. Para comprender mejor las técnicas, a continuación se describen algunas técnicas de codificación de la H.264/AVC, la codificación de vídeo de visualización múltiple desde la perspectiva de la ampliación H.264/MVC y la norma de Codificación de Vídeo de Alta Eficacia (HEVC) y técnicas de la 3D-AVC.

[0042] Para la norma H.264/Codificación Avanzada de Vídeo (AVC), la codificación o descodificación de vídeo (por ejemplo, la codificación) se implementa en macrobloques, donde un macrobloque representa una parte de una trama que tiene predicción inter o predicción intra (es decir, se codifica o descodifica por predicción inter o se codifica o descodifica por predicción intra). Por ejemplo, en la norma H.264/AVC, cada macrobloque (MB) interno (por ejemplo, macrobloque con predicción inter) se puede dividir en cuatro formas diferentes: una partición de MB 16x16, dos particiones de MB 16x8, dos particiones de 8x16 o cuatro particiones de MB 8x8. Las diferentes divisiones de MB de un MB pueden tener diferentes valores del índice de referencia para cada dirección (es decir, RefPicList0 o RefPicList1). Cuando un MB no se divide en múltiples (más de 1) particiones de MB, tiene solamente un vector de movimiento para toda la partición de MB en cada dirección.

[0043] Como parte de la codificación de vídeo (codificación o descodificación), el codificador de vídeo 30 puede configurarse para construir una o dos listas de imágenes de referencia, denominadas RefPicList0 y RefPicList1. La(s) lista(s) de imágenes de referencia identifica(n) imágenes de referencia que se pueden usar para realizar predicción inter de macrobloques de una trama o un fragmento. Por ejemplo, el codificador de vídeo 20 puede señalar un índice de referencia y un identificador de lista de imágenes de referencia. El descodificador de vídeo 30 puede recibir el índice de referencia y el identificador de la lista de imágenes de referencia y determinar la imagen de referencia que se va a usar para descodificar por predicción inter el macrobloque actual a partir del índice de referencia y del identificador de la lista de imágenes de referencia.

[0044] Cuando un MB se divide en cuatro particiones de MB 8x8, cada partición de MB 8x8 se puede dividir además en sub-bloques H.264/AVC. Existen cuatro formas diferentes de obtener sub-bloques H.264/AVC a partir de una partición de MB 8x8: un sub-bloque 8x8, dos sub-bloques 8x4, dos sub-bloques 4x8 o cuatro sub-bloques 4x4. Cada sub-bloque H.264/AVC puede tener un vector de movimiento diferente en cada dirección, pero compartir el mismo índice de imagen de referencia para cada dirección. La manera en que una partición de MB 8x8 se divide en sub-bloques se denomina partición en sub-bloques.

[0045] Esta divulgación generalmente utilizará el término bloque para referirse a cualquier bloque de datos de vídeo. Por ejemplo, en el contexto de la codificación según la norma H.264 y sus extensiones, un bloque puede referirse a cualquiera entre macrobloques, particiones de macrobloques, sub-bloques u otros tipos cualesquiera de bloques. En el contexto de la HEVC y sus extensiones, un bloque puede referirse a cualquiera de las PU, TU, CU u otros tipos cualesquiera de bloques. Un sub-bloque, como se usa en esta divulgación, generalmente se refiere a cualquier parte de un bloque más grande. Un sub-bloque mismo también puede ser mencionado simplemente como un bloque. Cuando se refiere específicamente a un sub-bloque como se define en H.264/AVC, se puede usar el término sub-bloque H.264/AVC. De lo contrario, el sub-bloque, como se usa en esta divulgación, generalmente pretende ser un término genérico que abarca el sub-bloque H.264/AVC, así como los otros tipos de sub-bloques descritos anteriormente.

[0046] Para la codificación de vídeo de visualización múltiple, existen múltiples normas diferentes de codificación de vídeo. Para evitar confusiones, cuando esta divulgación describe genéricamente la codificación de vídeo de visualización múltiple, esta divulgación usa la frase "codificación de vídeo de visualización múltiple". En general, en la codificación de vídeo de visualización múltiple, hay una visualización base y una o más visualizaciones no base o dependientes. La visualización base es totalmente descodificable sin referencia a ninguna de las visualizaciones dependientes (es decir, la visualización base solamente se tiene predicción inter con vectores de movimiento temporal). Esto permite que un códec que no está configurado para la codificación de vídeo de visualización múltiple reciba todavía al menos una visualización que sea totalmente descodificable (es decir, la visualización base se puede extraer y las otras visualizaciones se pueden descartar, permitiendo que un descodificador no configurado para la codificación de vídeo de visualización múltiple descodifique, sin embargo, el contenido del vídeo, aunque sea sin experiencia en 3D). Las una o más visualizaciones dependientes pueden tener predicción inter con respecto a la visualización base o con respecto a otra visualización dependiente (es decir, predecir con compensación de disparidad), o con respecto a otras imágenes en la misma visualización (es decir, predecir con compensación de movimiento).

[0047] Si bien la "codificación de vídeo de visualización múltiple" se usa genéricamente, el acrónimo MVC está asociado a una ampliación de la H.264/AVC. En consecuencia, cuando la divulgación usa el acrónimo MVC, la divulgación se refiere específicamente a la ampliación de la norma de codificación de vídeo H.264/AVC. La ampliación MVC de la H.264/AVC se basa en los vectores de movimiento de disparidad como otro tipo de vector de movimiento además de los vectores de movimiento temporal. Otra norma de codificación de vídeo, denominada MVC más profundidad (MVC+D), también ha sido desarrollada por JCT-3V y MPEG. MVC+D aplica las mismas herramientas de codificación de bajo nivel que las de MVC, tanto para la textura como para la profundidad, siendo la descodificación de la profundidad independiente de la descodificación de textura, y viceversa. Por ejemplo, en la MVC, una trama está representada solamente por un componente de visualización, denominado componente de visualización de textura o, simplemente, textura. En MVC+D, existen dos componentes de visualización: el componente de visualización de textura y el componente de visualización de profundidad o, simplemente, textura y profundidad. Por ejemplo, en MVC+D, cada visualización incluye una visualización de textura y una visualización de profundidad, donde la visualización incluye una pluralidad de componentes de visualización, la visualización de textura incluye una pluralidad de componentes de visualización de textura y la visualización de profundidad incluye una pluralidad de componentes de visualización de profundidad.

[0048] Cada componente de visualización de textura está asociado a un componente de visualización de profundidad para formar un componente de visualización de una visualización. El componente de visualización de profundidad representa la profundidad relativa de los objetos en el componente de visualización de textura. En MVC+D, el componente de visualización de profundidad y el componente de visualización de textura son descodificables por separado. Por ejemplo, el descodificador de vídeo 30 puede implementar dos casos de un códec de MVC, en los que un primer códec descodifica los componentes de visualización de textura y un segundo códec descodifica los componentes de visualización de profundidad. Estos dos códecs se pueden ejecutar de forma independiente uno del otro, porque los componentes de visualización de textura y los componentes de visualización de profundidad se codifican por separado.

[0049] En MVC+D, un componente de visualización de profundidad siempre sigue inmediatamente al componente de visualización de textura asociado (por ejemplo, correspondiente). De esta manera, la MVC+D presta soporte a la codificación de textura en primer lugar, donde el componente de visualización de textura se descodifica antes del componente de visualización de profundidad.

[0050] Un componente de visualización de textura y su componente de visualización de profundidad asociado (por ejemplo, correspondiente) pueden incluir el mismo valor de recuento de orden de imagen (POC) y el mismo id_vista (es decir, el valor de POC y el id_vista de un componente de visualización de textura y su componente asociado de visualización de profundidad son el mismo). El valor de POC indica el orden de visualización del componente de visualización de textura y el id_vista indica la visualización a la que pertenecen el componente de visualización de textura y el componente de visualización de profundidad.

[0051] La FIG. 2 muestra un típico orden de descodificación de la MVC (es decir, orden del flujo de bits). La disposición del orden de descodificación se denomina codificación de tiempo en primer lugar. Se debe tener en cuenta que el orden de descodificación de las unidades de acceso puede no ser idéntico al orden de salida o de visualización. En la FIG. 2, S0 a S7 se refieren, cada uno, a diferentes visualizaciones del vídeo de visualización múltiple. Cada una entre T0 a T8 representa una instancia de tiempo de salida. Una unidad de acceso puede incluir las imágenes codificadas de todas las visualizaciones para una instancia de tiempo de salida. Por ejemplo, una primera unidad de acceso puede incluir todas las visualizaciones S0 a S7 para la instancia de tiempo T0, una segunda unidad de acceso puede incluir todas las visualizaciones S0 a S7 para la instancia de tiempo T1 y así sucesivamente.

[0052] En aras de la brevedad, la divulgación puede utilizar las siguientes definiciones:

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

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

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

[0053] Como se ha expuesto anteriormente, en el contexto de esta divulgación, el componente de visualización, el componente de visualización de textura y el componente de visualización de profundidad pueden denominarse, en general, una capa. En la FIG. 2, cada una de las visualizaciones incluye conjuntos de imágenes. Por ejemplo, la visualización S0 incluye un conjunto de imágenes 0, 8, 16, 24, 32, 40, 48, 56 y 64, la visualización S1 incluye el conjunto de imágenes 1, 9, 17, 25, 33, 41, 49, 57 y 65, y así sucesivamente. Cada conjunto incluye dos imágenes: una imagen se denomina componente de visualización de textura y la otra imagen se denomina componente de visualización de profundidad. El componente de visualización de textura y el componente de visualización de profundidad dentro de un conjunto de imágenes de una visualización se pueden considerar como correspondientes entre sí. Por ejemplo, el componente de visualización de textura dentro de un conjunto de imágenes de una visualización se considera correspondiente al componente de visualización de profundidad dentro del conjunto de las imágenes de la visualización, y viceversa (es decir, el componente de visualización de profundidad corresponde a su componente de visualización de textura en el conjunto, y viceversa). Como se usa en esta divulgación, un componente de visualización de textura que corresponde a un componente de visualización de profundidad se puede considerar como el componente de visualización de textura, y el componente de visualización de profundidad como parte de una misma visualización de una única unidad de acceso.

[0054] El componente de visualización de textura incluye el contenido de imagen real que se muestra. Por ejemplo, el componente de visualización de textura puede incluir los componentes de luma (Y) y croma (Cb y Cr). El componente de visualización de profundidad puede indicar profundidades relativas de los píxeles en su correspondiente componente de visualización de textura. Como analogía a modo de ejemplo, el componente de visualización de profundidad es como una imagen en escala de grises que incluye solo valores de luma. En otras palabras, el componente de visualización de profundidad puede que no transmita ningún contenido de imagen, sino que, en cambio, proporcione una cierta medida de las profundidades relativas de los píxeles en el componente de visualización de textura.

[0055] Por ejemplo, un píxel blanco puro en el componente de visualización de profundidad indica que su correspondiente píxel, o correspondientes píxeles, en el correspondiente componente de visualización de textura, está(n) más cerca, desde la perspectiva del observador, y un píxel negro puro en el componente de visualización de profundidad indica que su correspondiente píxel, o correspondientes píxeles, en el correspondiente componente de visualización de textura, está(n) más lejos, desde la perspectiva del observador. Los diversos tonos de gris entre negro y blanco indican diferentes niveles de profundidad. Por ejemplo, un píxel muy gris en el componente de visualización de profundidad indica que su píxel correspondiente en el componente de visualización de textura está más alejado que un píxel ligeramente gris en el componente de visualización de profundidad. Dado que solamente es necesaria la escala de grises para identificar la profundidad de los píxeles, el componente de visualización de profundidad no necesita incluir componentes de croma, ya que los valores de color para el componente de visualización de profundidad pueden no tener ningún sentido. La explicación anterior pretende ser una analogía con fines de relacionar imágenes de profundidad con imágenes de textura. Los valores de profundidad en una imagen de profundidad, de hecho, no representan matices de gris pero, de hecho, representan valores de profundidad de 8 bits, o de otro tamaño en bits.

[0056] El componente de visualización de profundidad que utiliza solo valores de luma (por ejemplo, valores de intensidad) para identificar la profundidad se proporciona con fines ilustrativos y no debería considerarse limitativa. En otros ejemplos, se puede utilizar cualquier técnica para indicar las profundidades relativas de los píxeles en el componente de visualización de textura.

- 5 **[0057]** La FIG. 3 muestra una típica estructura de predicción de MVC (que incluye tanto la predicción entre imágenes dentro de cada visualización como la predicción entre visualizaciones) para la codificación de vídeo de visualización múltiple. Las direcciones de predicción se indican mediante flechas, utilizando, el objeto apuntado, el objeto desde el que se apunta como referencia de predicción. En la MVC, la predicción entre visualizaciones tiene soporte de la compensación de movimiento de disparidad, que usa la sintaxis de la compensación de movimiento de la H.264/AVC, pero permite que una imagen en una visualización diferente sea usada como imagen de referencia.
- 10 **[0058]** En el ejemplo de la FIG. 3, se ilustran ocho visualizaciones (que tienen identificadores de visualización "S0" a "S7") y se ilustran doce ubicaciones temporales ("T0" a "T11") para cada visualización. Es decir, cada fila en la FIG. 3 corresponde a una visualización, mientras que cada columna indica una ubicación temporal.
- 15 **[0059]** Aunque la MVC tiene una denominada visualización de base que es descodificable mediante los descodificadores de la H.264/AVC y los pares de visualizaciones estéreo podrían soportarse también por MVC, la ventaja de la MVC es que podría soportar un ejemplo que usa más de dos visualizaciones como una entrada de vídeo 3D y descodifica este vídeo 3D representado por las múltiples visualizaciones. Un representador de un cliente que tiene un descodificador de MVC puede esperar contenido de vídeo 3D con múltiples visualizaciones.
- 20 **[0060]** 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 parte del vídeo. Esta divulgación puede usar los términos imagen y trama de forma intercambiable.
- 25 **[0061]** Las imágenes en la FIG. 3 se ilustran usando un bloque que incluye una letra, designando la letra si la imagen correspondiente está intra-codificada (es decir, es una imagen I), o inter-codificada en una dirección (es decir, como una imagen P) o en múltiples direcciones (es decir, como una imagen B). En general, las predicciones se indican mediante flechas, donde las imágenes a las que se apunta usan la imagen desde la que se apunta como referencia de predicción. Por ejemplo, la imagen P de la visualización S2 en la ubicación temporal T0 se predice a partir de la imagen I de la visualización S0 en la ubicación temporal T0.
- 30 **[0062]** Al igual que con la codificación de vídeo de visualización única, las imágenes de una secuencia de vídeo de codificación de vídeo de múltiples visualizaciones pueden codificarse predictivamente con respecto a imágenes en diferentes ubicaciones temporales. Por ejemplo, la imagen b de la visualización S0 en la ubicación temporal T1 tiene una flecha apuntando a la misma desde la imagen I de la visualización S0 en la ubicación temporal T0, indicando que la imagen b se predice a partir de la imagen I. Adicionalmente, sin embargo, en el contexto de la codificación de vídeo de múltiples visualizaciones, las imágenes se pueden predecir entre visualizaciones. Es decir, un componente de visualización puede usar los componentes de visualización en otras visualizaciones como referencia. En la MVC, por ejemplo, la predicción entre visualizaciones se realiza como si el componente de visualización de otra visualización fuera una referencia de predicción inter. Las posibles referencias entre visualizaciones se señalizan en la extensión de la MVC del conjunto de parámetros de secuencia (SPS) y pueden ser modificadas por el proceso de construcción de la lista de imágenes de referencia, que habilita el ordenamiento flexible de las referencias de predicción inter o de predicción entre visualizaciones. La predicción entre visualizaciones también es una característica de la ampliación propuesta de visualización múltiple de la HEVC, incluyendo la 3D-HEVC (visualización múltiple más profundidad).
- 35 **[0063]** La FIG. 3 proporciona varios ejemplos de predicción entre visualizaciones. Las imágenes de la visualización S1, en el ejemplo de la FIG. 3, se ilustran como predichas a partir de imágenes en diferentes ubicaciones temporales de la visualización S1, así como predichas entre visualizaciones a partir de imágenes de las visualizaciones S0 y S2 en las mismas ubicaciones temporales. Por ejemplo, la imagen b de la visualización S1 en la ubicación temporal T1 se predice a partir de cada una de las imágenes B de la visualización S1 en las ubicaciones temporales T0 y T2, así como las imágenes b de las visualizaciones S0 y S2 en la ubicación temporal T1.
- 40 **[0064]** En algunos ejemplos, la FIG. 3 puede verse como una ilustración de los componentes de la visualización de textura. Por ejemplo, las imágenes I, P, B y b, ilustradas en la FIG. 2, se pueden considerar componentes de visualización de textura para cada una de las visualizaciones. de acuerdo con las técnicas descritas en esta divulgación, para cada uno de los componentes de visualización de textura ilustrados en la FIG. 3 existe un correspondiente componente de visualización de profundidad. En algunos ejemplos, los componentes de visualización de profundidad se pueden predecir de una manera similar a la ilustrada en la FIG. 3 para los correspondientes componentes de visualización de textura.
- 45 **[0065]** La codificación de dos visualizaciones también puede tener soporte de la MVC. Una de las ventajas de la MVC es que un codificador de MVC puede tomar más de dos visualizaciones como una entrada de vídeo 3D y un descodificador de MVC puede descodificar dicha representación de visualización múltiple. De tal modo, cualquier representador con un descodificador de MVC puede descodificar contenidos de vídeo 3D con más de dos visualizaciones.
- 50 **[0066]** Como se ha expuesto anteriormente, en la MVC, se permite la predicción entre visualizaciones entre imágenes en la misma unidad de acceso (lo que quiere decir, en algunos casos, con la misma instancia de tiempo).
- 55
- 60
- 65

Cuando se codifica una imagen en una de las visualizaciones no básicas, se puede añadir una imagen a una lista de imágenes de referencia, si está en una visualización diferente pero dentro de una misma instancia de tiempo. Una imagen de referencia de predicción entre visualizaciones se puede poner 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 visualización puede usar los componentes de visualización en otras visualizaciones como referencia. En la MVC, la predicción entre visualizaciones se realiza como si el componente de visualización en otra visualización fuera una referencia de predicción inter.

[0067] En la MVC, se permite la predicción entre visualizaciones 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 visualizaciones no básicas, se puede añadir una imagen a una lista de imágenes de referencia si está en una visualización diferente pero con una misma instancia de tiempo. Una imagen de referencia de predicción entre visualizaciones se puede poner en cualquier posición de una lista de imágenes de referencia, tal como cualquier imagen de referencia de predicción inter.

[0068] Como se muestra en la FIG. 3, un componente de visualización puede usar los componentes de visualización en otras visualizaciones como referencia. Esto se llama predicción entre visualizaciones. En la MVC, la predicción entre visualizaciones se realiza como si el componente de visualización en otra visualización fuera una referencia de predicción inter.

[0069] En el contexto de la codificación de vídeo de múltiples visualizaciones, hay dos tipos de vectores de movimiento: uno es un vector de movimiento normal que apunta a imágenes de referencia temporales. La predicción inter temporal correspondiente es la predicción compensada por movimiento (MCP). El otro tipo de vector de movimiento es un vector de movimiento de disparidad que apunta a imágenes en una visualización diferente (es decir, imágenes de referencia entre visualizaciones). La predicción inter correspondiente es la predicción compensada por disparidad (DCP).

[0070] La siguiente sección analizará la norma de codificación de vídeo 3D basado en AVC. En la actualidad, un Equipo de Colaboración Conjunta en Codificación de Vídeo 3D (JCT-3V) de VCEG y MPEG está desarrollando una norma 3DV basada en H.264/AVC, es decir, 3D-AVC. Para 3D-AVC, se han incluido y admitido nuevas herramientas de codificación además de la predicción entre visualizaciones en MVC. El software 3D-ATM más reciente para la 3D-AVC se puede descargar desde el siguiente enlace: [3D-ATM versión 6,2]: <http://mpeg3dv.research.nokia.com/svn/mpeg3dv/tags/3DV-ATMv6,2/>

[0071] La norma de codificación de vídeo 3D basada en AVC (3D-AVC) está actualmente en desarrollo por JCT-3V, y la versión más reciente de 3D-AVC ahora está disponible al público: MM Hannuksela, Y. Chen, T. Suzuki, J.-R. Ohm, G.J. Sullivan, "3D-AVC draft text 5 [Borrador de texto de 3D-AVC 5]", JCT3V-C1002, Ginebra, Suiza, enero de 2013. Está disponible, a partir del 14 de marzo de 2014, en el siguiente enlace.

[0072] Ahora se analizará el orden de codificación de los componentes de visualización en 3D-AVC. 3D-AVC es compatible con la norma H.264/AVC de tal forma que la parte de textura de la visualización básica es completamente descodificable mediante un descodificador H.264/AVC. Por ejemplo, los componentes de visualización de textura en los componentes de visualización de la visualización básica solamente pueden tener predicción inter con otros componentes de visualización de textura en la misma visualización básica. Es posible que los componentes de visualización de textura en la visualización básica no tengan predicción entre visualizaciones. También, es posible que el componente de visualización de textura en la visualización básica no requiera el componente de visualización de profundidad correspondiente para propósitos de descodificación.

[0073] Para los componentes visualización mejorados en 3D-AVC, en algunas otras técnicas de ejemplo, la profundidad se puede codificar antes de la textura y un componente de visualización de textura se puede codificar basándose en la información del componente de visualización de profundidad, que también se conoce como codificación de profundidad primero. Sin embargo, cada componente de visualización de textura se codifica antes que los componentes de visualización de profundidad respectivos en el orden de codificación de textura primero, tal como en MVC+D descrita anteriormente. En otras palabras, en algunas otras técnicas de ejemplo, en 3D-AVC, el componente de visualización de textura de la visualización básica se codifica primero, seguido del componente de visualización de profundidad asociado de la visualización básica, seguido del componente de visualización de profundidad de una primera visualización de mejora o dependiente, seguido del componente de visualización de textura asociado de la primera visualización de mejora o dependiente, seguido del componente de visualización de profundidad de una segunda visualización de mejora o dependiente, seguido del componente de visualización de textura asociado de la segunda visualización de mejora o dependiente, y así sucesivamente.

[0074] Por ejemplo, las órdenes de codificación de los componentes de visualización de textura y profundidad en 3D-AVC se ejemplifican de la forma siguiente. En los siguientes ejemplos, T0 y D0, respectivamente, se refieren a los componentes de visualización de textura y profundidad de la visualización básica, y Ti y Di, respectivamente, se refieren a los componentes de visualización de textura y profundidad de la i-ésima visualización dependiente. En los siguientes ejemplos, se consideran tres visualizaciones.

[0075] En un primer ejemplo, las visualizaciones consideradas son T0, D0, D1, D2, T1 y T2. En este ejemplo, las visualizaciones básicas (T0 y D0) se codifican con el orden de codificación de textura primero, mientras que las visualizaciones dependientes se codifican con el orden de codificación de profundidad primero. Actualmente se usa un orden de codificación híbrido en condiciones de prueba comunes de 3D-AVC. En otro ejemplo, el orden de codificación es T0, D0, T1, D1, T2 y D2. Es decir, todos los componentes de visualización se codifican con el orden de codificación de textura primero.

[0076] Si la predicción entre visualizaciones está habilitada para T1, la visualización de textura de referencia se define como la visualización que incluye la imagen de referencia entre visualizaciones, y la visualización de profundidad correspondiente se define como la visualización de profundidad de referencia que tiene el mismo índice de orden de visualización que la de la visualización de textura de referencia.

[0077] Ahora se analizará la obtención de vectores de disparidad 3D-AVC a través del mapa de profundidad. Las técnicas para obtener el vector de disparidad pueden variar con cada herramienta de codificación de bajo nivel, pero, comúnmente, los datos de profundidad de la visualización dependiente se usan para la obtención de vectores de disparidad para la codificación del componente de visualización de textura. Esto es porque la visualización de profundidad de la visualización dependiente está disponible, debido al orden de codificación de profundidad primero. Las herramientas de codificación de bajo nivel usadas son la predicción entre visualizaciones de síntesis de visualización basada en bloques en bucle (BVSP) y la predicción de vectores de movimiento basada en profundidad (D-MVP) en 3D-AVC. Un codificador de vídeo, por ejemplo, el descodificador de vídeo 30 puede usar el vector de disparidad convertido a partir de los valores de profundidad de la visualización de profundidad (a veces llamado mapa de profundidad) en la visualización dependiente (a veces llamada trama dependiente). En el software de referencia 3D-AVC, típicamente, los resultados del proceso de conversión del valor del mapa de profundidad real a una disparidad con una visualización particular se almacenan en tablas de consulta con parámetros de cámara.

[0078] Ahora se analizará la obtención de profundidad máxima desde cuatro esquinas, para el uso en la obtención de vector de disparidad. Para obtener el valor de profundidad, el descodificador de vídeo 30 primero identifica el bloque de profundidad de referencia de un componente de visualización de profundidad. El bloque de profundidad de referencia esta co-ubicado/corresponde al MB/partición/sub-bloque actual. En el bloque de profundidad de referencia identificado, el descodificador de vídeo 30 accede a las cuatro muestras de esquina correspondientes a las muestras de profundidad superior izquierda, superior derecha, inferior izquierda e inferior derecha. A continuación se calcula un valor de profundidad tomando el máximo de las cuatro muestras de profundidad de esquina.

[0079] Finalmente, el descodificador de vídeo 30 infiere el componente horizontal de un vector de disparidad de la tabla de consulta utilizando el valor de profundidad calculado y estableciendo el componente vertical del vector de disparidad en 0. En este procedimiento, el número de muestras de profundidad a las que se accede aumenta cuando el MB se divide en particiones o sub-bloques. Por ejemplo, cuando un MB 16x16 se divide en cuatro particiones 8x8, el número de muestras de profundidad a las que se debe acceder es 16, y cuando un MB 16x16 se divide en dieciséis particiones 4x4, el número de las muestras de profundidad a las que se debe acceder es 64.

[0080] BVSP en 3D-AVC es un modo de codificación compatible con 3D-AVC, así como otras normas de codificación. BVSP se propuso originalmente en "3DV-CE1,a: Block-Based View Synthesis Prediction for 3DV-ATM [3DV-CE1,a: Predicción de síntesis de visualización basada en bloques para 3DV-ATM]" (JCT3V-A0107) por W. Su, *et al.*, que se puede descargar desde el siguiente enlace: http://phenix.it-sudparis.eu/jct2/doc_end_user/documents/1_Stockholm/wg11/JCT3V-A0107-v1.zip.

[0081] La FIG. 4 es un diagrama conceptual de la BVSP basándose en la distorsión inversa. Haciendo referencia a la FIG. 4, suponga que se utiliza el siguiente orden de codificación, (T0, D0, D1, T1), con T que se refiere a las visualizaciones de textura y D que se refiere a las visualizaciones de profundidad. El componente de textura T0 es una visualización básica, y T1 es una visualización dependiente codificada con VSP. Los componentes de mapa de profundidad D0 (no mostrados en la FIG. 4) y D1 son mapas de profundidad respectivos asociados con T0 y T1.

[0082] En la visualización dependiente T1, los valores de muestra del bloque codificado actualmente Cb se predicen a partir del área de referencia R(Cb), utilizando predicción VSP que consiste en los valores de muestra de la visualización básica T0. El vector de desplazamiento (Disp_vec) entre las muestras actuales que se van a codificar (es decir, Cb) y las muestras de referencia (es decir, R(Cb)) se indica como un vector de disparidad obtenido entre T1 y T0 a partir de un valor de mapa de profundidad asociado con una muestra de textura codificada actualmente.

[0083] El proceso de conversión de un valor de profundidad a un vector de disparidad se puede realizar por ejemplo con las siguientes ecuaciones:

$$Z(Cb(j,i)) = \frac{1}{\frac{d(Cb(j,i))}{255} \cdot \left(\frac{1}{Z_{near}} - \frac{1}{Z_{far}} \right) + \frac{1}{Z_{far}}}; \quad (1)$$

$$D(Cb(j,i)) = \frac{f \cdot b}{Z(Cb(j,i))}; \quad (2)$$

donde j e i son coordenadas espaciales locales dentro de Cb , $d(Cb(j,i))$ es un valor de mapa de profundidad en la imagen de mapa de profundidad de una visualización 1, Z es el valor de profundidad correspondiente real y D es el componente horizontal de un vector de disparidad obtenido a una visualización particular 0. Los parámetros f , b , Z_{cerca} y Z_{lejos} son parámetros que especifican la configuración de la cámara, es decir, la distancia focal usada (f), la separación de la cámara (b) entre la visualización n.º 1 y la visualización n.º 0, y el intervalo de profundidad (Z_{cerca} , Z_{lejos}) representan parámetros de conversión del mapa de profundidad.

[0084] Se debe tener en cuenta que, en algunos ejemplos, el componente vertical del vector de disparidad obtenido se establece en 0. También, en algunas implementaciones de 3DV-ATM, las ecuaciones (1) y (2) ya se han calculado previamente para cada valor del mapa de profundidad (0..255) y se almacenaron como una tabla de consulta.

[0085] En el ejemplo de la FIG. 3, Cb representa un bloque actualmente codificado. El descodificador de vídeo 30 puede codificar Cb usando un modo BVSP. Si Cb se va a codificar utilizando el modo BVSP, el descodificador de vídeo 30 identifica el bloque de profundidad, $d(Cb)$ en la FIG. 4, correspondiente a Cb . En este sentido, correspondiente significa co-ubicado. El bloque de profundidad $d(Cb)$ incluye una pluralidad de valores de profundidad. Basándose en uno o más de esos valores de profundidad de $d(Cb)$, el descodificador de vídeo 30 determina un valor de disparidad. Ese valor de disparidad se puede utilizar como el componente x de un vector de disparidad, con el componente y establecido en 0. Usando el vector de disparidad, el descodificador de vídeo 30 puede identificar un bloque de referencia, $R(cb)$ en el ejemplo de la FIG. 4, en una visualización diferente y realizar predicción inter del bloque Cb basándose en ese bloque de referencia.

[0086] La resolución de la visualización de profundidad y la resolución de la visualización de textura pueden ser iguales, o pueden ser diferentes. Si la resolución de la visualización de profundidad y las resoluciones de la visualización de textura son diferentes, entonces el descodificador de vídeo 30 puede realizar una conversión para encontrar la profundidad co-ubicada del bloque. El descodificador de vídeo 30 puede realizar la conversión de la siguiente manera. Sea (x, y) la posición superior izquierda del bloque Cb . La posición de la esquina superior izquierda del bloque co-ubicado en la visualización de profundidad se representa mediante $(x \gg \text{reduced_resolution_flag}, y \gg \text{reduced_resolution_flag})$. El elemento sintáctico "reduced_resolution_flag" es igual a 1 para especificar que los componentes de visualización de profundidad de un par de componentes de visualización tienen una resolución espacial más baja que el componente de luma del componente de visualización de textura del mismo par de componentes de visualización, y tanto la anchura como la altura de los componentes de visualización de profundidad son la mitad de la anchura y la altura de todos los componentes de visualización de textura.

[0087] La siguiente sección de esta divulgación analizará varios problemas de implementación de BVSP. Un problema involucra la indicación de bloques BVSP, es decir, qué bloques deben codificarse utilizando el modo BVSP. Los bloques BVSP se indican de la siguiente manera:

- Se usa un indicador en el nivel de MB para señalar si el MB actual se codifica con el modo de salto/directo convencional o si se codifica con el modo de salto/directo pero se predice a partir de un componente de referencia sintético. En este contexto, "el modo de salto/directo convencional se refiere a la versión extendida del modo de salto/directo utilizado en la norma de codificación H.264/AVC, y un componente de referencia sintético se refiere a un bloque de referencia generado a partir de un bloque entre visualizaciones.

- Para cada división de MB (de 16x16 a 8x8), se usa un índice de referencia (o un indicador, como en algunas propuestas para 3D-AVC) en cada lista de imágenes de referencia para señalar la imagen de referencia. Cuando se codifica una división en el modo BVSP, las diferencias del vector de movimiento no se señalan, ya que no existen vectores de movimiento para los bloques codificados BVSP.

[0088] Cuando el indicador o bien el índice de referencia indica un componente de referencia sintético, se aplica la predicción de una división como se describe en el siguiente punto. En otras palabras, un bit puede indicar si la partición MB o MB está codificada utilizando los modos de salto/directo convencionales o un modo BVSP. En el modo BVSP, el descodificador de vídeo 30 descodifica la parte de una partición MB o MB que se codifica utilizando el modo BVSP (usted está llamando como salto/dirección sintético) al dividir la partición MB o MB en bloques $K \times K$, y para cada bloque $K \times K$, identificando el bloque de profundidad colocado, obteniendo un valor de disparidad del bloque de profundidad, y prediciendo el MB o la partición de MB a partir de la imagen de referencia entre visualizaciones apuntada por el vector de disparidad.

[0089] En 3D-AVC, para el modo de salto/directo convencional, el descodificador de vídeo 30 obtiene el vector de movimiento y el índice de referencia de la siguiente manera. El vector de movimiento del bloque correspondiente (señalado por el vector de disparidad) en la visualización de referencia, si está disponible, se establece igual al vector de movimiento del bloque actual, y el índice de referencia del bloque actual se obtiene en consecuencia. Si el vector de movimiento entre visualizaciones no está disponible, es decir, el bloque de referencia en la visualización base

apuntada por el vector de disparidad está intracodificado, se utiliza un esquema de predicción de vectores de movimiento basado en la mediana convencional, al establecer el índice de referencia en cero.

5 **[0090]** Otra cuestión de implementación relacionada con la BVSP implica el proceso de obtención de predicción. Para cada división de MB, con su tamaño indicado por $N \times M$ (en el que N o M serán 8 o 16), si la división de MB se codifica con el modo BVSP, una división de MB actual se divide adicionalmente en varias sub-regiones con un tamaño igual a $K \times K$ (en el que K puede ser 8×8 , como en algunas propuestas para 3D-AVC, 4×4 , 2×2 o 1×1). Para cada sub-región, se obtiene un vector de disparidad separado y cada sub-región se predice a partir de un bloque ubicado mediante el vector de disparidad obtenido en la imagen de referencia entre visualizaciones, por ejemplo, $R(cb)$ en la FIG. 4. En algunas condiciones de prueba comunes de ejemplo, K se define como 4. Se debe tener en cuenta que los vectores de disparidad obtenidos no se almacenan para los bloques codificados BVSP, ya que no existe ninguna herramienta de codificación que use dichos vectores.

15 **[0091]** Otro problema de implementación con BVSP involucra el proceso de obtención de vector de disparidad. Cuando se aplica un primer orden de codificación de profundidad, el vector de disparidad se obtiene al convertir un valor de profundidad del bloque de profundidad correspondiente en la visualización de profundidad no básica correspondiente, como se muestra en la FIG. 4. Se calcula un solo valor de profundidad accediendo primero a las cuatro muestras de profundidad de esquina desde el bloque de profundidad co-ubicado/correspondiente a la sub-región $K \times K$ y luego tomando el valor máximo de las cuatro muestras de profundidad a las que se accede. El valor de profundidad calculado se convierte más tarde en un vector de disparidad utilizando las ecuaciones (1) y (2). Cuando se aplica el primer orden de codificación de textura, los modos de BVSP se inhabilitan, ya que la visualización de profundidad no básica correspondiente no está disponible cuando se descodifica la visualización de textura no básica.

25 **[0092]** A continuación se analizará la predicción de vectores de movimiento basada en profundidad (D-MVP) en 3D-AVC para los modos de predicción inter normales. D-MVP se refiere a un procedimiento de predicción de vectores de movimiento que utiliza los datos de mapa de profundidad asociados en la visualización actual, que están disponibles debido al primer orden de codificación de profundidad. El procedimiento se aplica con los componentes de visualización de textura en visualizaciones dependientes.

30 **[0093]** En 3D-AVC, la predicción de vectores de movimiento todavía utiliza los bloques contiguos. Los bloques contiguos incluyen, por orden, un bloque izquierdo, un bloque superior, un bloque superior derecho y un bloque superior izquierdo del bloque actual. El vector de movimiento en el bloque superior izquierdo se utiliza solo cuando uno de los otros tres bloques contiguos no contenga un vector de movimiento y, por tanto, se considera que no está disponible.

35 **[0094]** El vector de movimiento de un bloque contiguo se considera no disponible si tiene un tipo diferente como el vector de movimiento actual que debe predecirse. El tipo de un vector de movimiento depende del índice de referencia correspondiente. Es decir, si el índice de referencia corresponde a una imagen de referencia entre visualizaciones, el vector de movimiento es un vector de movimiento de disparidad y el tipo es "disparidad" y si el índice de referencia corresponde a una imagen de referencia temporal (en la misma visualización), el vector de movimiento es un vector de movimiento temporal y el tipo es "temporal".

45 **[0095]** En 3D-AVC, si los tres bloques contiguos están disponibles, los vectores de movimiento en los tres bloques contiguos se emplean para la predicción de vectores de movimiento del bloque actual. En la predicción temporal, si todos sus vectores de movimiento tienen el mismo tipo y tienen los mismos índices de referencia, el filtro de mediana se usa directamente como en H.264/AVC, de lo contrario (si pertenecen a tipos diferentes y tienen índices de referencia diferentes), se obtiene adicionalmente un movimiento vector. Cuando la imagen de referencia actual es una imagen de referencia entre visualizaciones, se comprueban los tipos de vectores de movimiento y sus índices de referencia en las posiciones de bloques contiguos y, si tienen todos el mismo tipo y los mismos índices de referencia, se aplica el filtro de mediana. En ambos casos, si están disponibles menos de tres bloques contiguos, los vectores de movimiento para los bloques no disponibles se obtienen adicionalmente para que tres bloques contiguos se vuelvan disponibles.

55 **[0096]** Si un bloque contiguo espacial no contiene un vector de movimiento temporal disponible, se debe predecir un vector de movimiento temporal para el bloque actual. Un bloque de referencia del bloque actual en la imagen de visualización de referencia se identifica mediante el vector de disparidad obtenido como se describe anteriormente con respecto a la predicción de la síntesis de visualización basada en bloques. El vector de movimiento de un bloque que contiene la posición central del bloque de referencia, si es un vector de movimiento temporal, se obtiene para el bloque contiguo espacial actual. Si se considera que el vector de movimiento temporal no está disponible (bloque intra o no apuntando a una imagen de referencia en la visualización de referencia alineada con la imagen de referencia actual), el vector de movimiento obtenido se establece en cero.

60 **[0097]** Si un bloque contiguo espacial no contiene un vector de movimiento de disparidad disponible y se debe predecir un vector de movimiento de disparidad para el bloque actual, el vector de disparidad obtenido para el bloque actual se convierte en un vector de movimiento de disparidad como se describió anteriormente con respecto a la predicción de síntesis de visualización basada en bloques.

65

[0098] En 3D-AVC, el procedimiento D-MVP se incorpora a la predicción de vectores de movimiento basada en la función mediana convencional e H.264/AVC. Por lo tanto, cuando los vectores de movimiento en bloques contiguos espaciales están disponibles (o no están disponibles originalmente pero están disponibles con los procedimientos mencionados anteriormente), la función de la mediana se puede aplicar aún a tres vectores de movimiento; sin embargo, todos deben pertenecer al mismo tipo.

[0099] A continuación se analizará la predicción de movimiento entre visualizaciones en 3D-AVC para los modos de salto y directo. La predicción de movimiento entre visualizaciones en 3D-AVC se realiza en salto de P, salto de B, modo directo de B 16x16 y modos directos de B 8x8. En estos modos, un vector de disparidad utilizado para identificar un bloque de referencia proviene de los bloques contiguos o del vector de disparidad obtenido (ver más arriba en relación con la predicción de la síntesis de visualización basada en bloques) del bloque de profundidad correspondiente asociado con el MB actual. Los bloques contiguos candidatos A, B y C se comprueban para determinar si están disponibles o no con un vector de movimiento de disparidad. Si está disponible un bloque contiguo espacial, es decir, y contiene un vector de movimiento de disparidad, este vector de movimiento de disparidad se convierte en el vector de disparidad.

[0100] Si el vector de disparidad para los bloques contiguos espaciales A, B y C no está disponible, se usa un vector de disparidad obtenido del bloque de profundidad (ver más arriba en relación con la predicción de la síntesis basada en bloques) asociado con el MB actual para el bloque no disponible. Posteriormente, se aplica un filtro de mediana para obtener el vector de disparidad.

[0101] Este vector de disparidad obtenido del proceso anterior se utiliza para obtener un bloque de referencia en una imagen de visualización de referencia. Dentro del bloque de referencia, el vector de movimiento (es decir, el vector de movimiento entre visualizaciones) si está disponible, se establece igual al vector de movimiento del bloque actual y el índice de referencia del bloque actual se obtiene en consecuencia.

[0102] Si el vector de movimiento entre visualizaciones no está disponible, es decir, el bloque de referencia en la visualización base apuntada por el vector de disparidad está intracodificado, se utiliza el esquema de predicción de vectores de movimiento basado en la mediana convencional. En este caso, primero se obtiene un índice de referencia y el esquema D-MVP descrito en la sección 1.4.5 se usa para obtener la predicción del vector de movimiento para el MB actual.

[0103] Ahora se analizarán las mejoras para la obtención de vector de disparidad. En JCT3V-C0122, se ha propuesto un procedimiento de obtención de vector de disparidad simplificado. En el procedimiento propuesto, cuando el MB actual se codifica con modos de predicción inter que no se saltan ni son directos, todos los particiones/sub-bloques con el MB actual comparten el vector de disparidad obtenido que se calcula a partir de la muestra de profundidad inferior derecha del bloque de profundidad asociado del MB actual. Sin embargo, cuando el MB actual se codifica con un modo de salto o directo, se utiliza un proceso de obtención de vector de disparidad diferente, es decir, el acceso a cuatro muestras de esquina del bloque de profundidad de referencia. Además, en el modo BVSP, aún es necesario acceder a las cuatro muestras de esquina del bloque de profundidad de referencia de cada sub-región KxK de una partición MB.

[0104] En JCT3V-C0134, se propone que cuando se codifican con modos inter, todos los bloques de partición dentro del mismo MB comparten el único vector de disparidad que se obtiene de un valor de profundidad máximo de cuatro muestras de esquina del mismo bloque de profundidad de referencia que está co-ubicado/correspondiente al MB actual. Sin embargo, para el modo BVSP, aún es necesario acceder a las cuatro muestras de esquina del bloque de profundidad de referencia de cada sub-región KxK de una partición MB.

[0105] Esta divulgación ahora analizará aspectos de 3D-HEVC. La obtención de vector de disparidad basado en bloques contiguos (NBDV, por sus siglas en inglés) se puede usar como un procedimiento de obtención de vector de disparidad en 3D-HEVC que usa el primer orden de codificación de textura para todas las visualizaciones. En el diseño actual de 3D-HEVC, el vector de disparidad obtenido a partir del NBDV se puede refinar aún más recuperando los datos de profundidad del mapa de profundidades de la visualización de referencia. En NBDV, se usa un vector de disparidad (DV) como estimador del desplazamiento entre dos visualizaciones. 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 en bloques contiguos como un buen predictor. Siguiendo esta idea, la NBDV utiliza la información de disparidad contigua para estimar el vector de disparidad en diferentes visualizaciones.

[0106] Inicialmente se definen varios bloques contiguos en el espacio y en el tiempo. A continuación, cada uno de los bloques contiguos se comprueba en un orden predefinido determinado por la prioridad de la correlación entre el bloque actual y el bloque candidato. Una vez que un vector de movimiento de disparidad (es decir, el vector de movimiento que apunta a una imagen de referencia entre visualizaciones) se encuentra en los candidatos, el vector de movimiento de disparidad se convierte en un vector de disparidad. Se utilizan dos conjuntos de bloques contiguos. Un conjunto es de bloques contiguos espacialmente y el otro conjunto es de bloques contiguos temporalmente.

[0107] 3D-HEVC adoptó primero el procedimiento de vector de disparidad de (basado en) bloques contiguos (NBDV) propuesto en JCT3V-A0097, 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 una NBDV simplificada en JCTVC-A0126, 3D-CE5.h: Simplificación de la obtención de vectores de disparidad para codificación de vídeo 3D basada en HEVC, J. Sung, M. Koo, S. Yea (LG). Además de eso, en JCT3V-B0047, 3D-CE5,h relacionado: Improvements for disparity vector derivation [Mejoras en la obtención de vector de disparidad], J. Kang, Y. Chen, L. Zhang, M. Karczewicz (Qualcomm), el NBDV se simplifica aún más al eliminar los vectores de disparidad implícitos almacenados en la memoria intermedia de imágenes descodificadas, pero también mejoró una ganancia de codificación con la selección de imágenes RAP.

[0108] En el NBDV actual (a la fecha de esta divulgación), se usan cinco bloques contiguos espaciales para la obtención de vectores de disparidad. Son los bloques inferior izquierdo, izquierdo, superior derecho, superior y superior izquierdo del bloque actual (por ejemplo, unidad de predicción actual (PU)), según lo indicado por A0, A1, B0, B1 o B2 en la FIG. 5. Cabe destacar que estos bloques contiguos son los mismos que los utilizados en los modos de FUSIÓN en HEVC. Por lo tanto, no se requiere ningún acceso adicional a la memoria.

[0109] Para la comprobación de bloques contiguos en el tiempo, un codificador de vídeo puede realizar primero un proceso de confección de una lista de imágenes candidatas. En un ejemplo, hasta dos imágenes de referencia provenientes de la visualización actual se pueden tratar como imágenes candidatas. El codificador de vídeo puede insertar primero la imagen de referencia co-ubicada en la lista de imágenes candidatas, seguida por el resto de imágenes candidatas en el orden ascendente del índice de imagen de referencia. Cuando las imágenes de referencia con el mismo índice de referencia en ambas listas de imágenes de referencia están disponibles, la que está en la misma lista de imágenes de referencia de la imagen de co-situada precede a la otra. Para cada imagen candidata en la lista de imágenes candidatas, se pueden determinar tres regiones candidatas para obtener los bloques de contiguidad temporal.

[0110] Cuando se codifica un bloque con predicción de movimiento entre visualizaciones, se puede obtener un vector de disparidad para seleccionar un bloque correspondiente en una visualización diferente. Un vector de disparidad implícito (IDV, también conocido como vector de disparidad obtenido) se refiere al vector de disparidad obtenido en la predicción de movimiento entre visualizaciones. Incluso aunque el bloque esté codificado con predicción de movimiento, el vector de disparidad obtenido no se descarta para el fin de codificar un bloque siguiente.

[0111] En el diseño actual de 3D-HTM 6,0, el codificador de vídeo, de acuerdo con el proceso NBDV, comprueba los vectores de movimiento de disparidad en los bloques contiguos temporales, los vectores de movimiento de disparidad en los bloques contiguos espaciales y luego los IDV en orden. Una vez que se encuentra el vector de movimiento de disparidad, o IDV, se termina el proceso.

[0112] Un codificador de vídeo puede refinar aún más los vectores de disparidad obtenidos usando NBDV mediante el acceso a información de profundidad. Cuando un vector de disparidad se obtiene del proceso NBDV, el codificador de vídeo puede refinar aún más el vector de disparidad al recuperar los datos de profundidad del mapa de profundidad de la visualización de referencia. El proceso de refinamiento puede incluir dos pasos:

a) Ubicar un bloque de profundidad correspondiente mediante el vector de disparidad obtenido en la visualización de profundidad de referencia previamente codificada, tal como la visualización de base; 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 correspondiente bloque de profundidad y convertirlo en el componente horizontal del vector de disparidad refinado. El componente vertical del vector de disparidad no se modifica.

[0113] El codificador de vídeo puede usar el vector de disparidad refinado para la predicción de movimiento entre visualizaciones mientras que el vector de disparidad no refinado se usa para la predicción residual entre visualizaciones. Además, el vector de disparidad refinado puede almacenarse como el vector de movimiento de una PU si se codifica con el modo VSP hacia atrás.

[0114] De acuerdo con las técnicas de esta divulgación, uno de los bloques espacialmente contiguos puede corresponder a un bloque codificado con BVSP, y otro de los bloques espacialmente contiguos puede corresponder a un bloque no codificado con BVSP. Por ejemplo, el bloque A1 puede corresponder a un bloque codificado con BVSP y el bloque B1 puede corresponder a un bloque no codificado con BVSP. Sin embargo, al codificar la información de movimiento para el bloque actual, un codificador de vídeo puede acceder a la información de movimiento tanto para el bloque A1 como para el bloque B1 utilizando la misma función lógica. Se presume que la información de movimiento para el bloque codificado con BVSP, es decir, el bloque A1 en el ejemplo mencionado anteriormente, incluye un índice de referencia que identifica una imagen de referencia. Por lo tanto, no es necesario proporcionar una función lógica separada para acceder a la información de movimiento del bloque A1 en el codificador de vídeo.

[0115] La FIG. 6 es un diagrama conceptual que ilustra técnicas relacionadas con la predicción de síntesis de visualización hacia atrás (BVSP) utilizando bloques contiguos. Se ha propuesto y adoptado BVSP como técnica para 3D-HEVC. El enfoque de VSP de distorsión inversa, según lo propuesto en JCT3V-C0152, fue adoptado en la tercera reunión del JCT-3V. La idea básica de esta VSP de distorsión inversa es la misma que la VSP basada en bloques en 3D-AVC. Ambas técnicas usan la distorsión inversa y la VSP basada en bloques para evitar transmitir las diferencias de vectores de movimiento y utilizar vectores de movimiento más precisos. Los detalles de implementación son diferentes debido a las diferentes plataformas. Esta divulgación generalmente utiliza el inicialismo BVSP para referirse a la predicción de síntesis de visualización hacia atrás en 3D-HEVC, aunque BVSP también puede referirse a la predicción de síntesis de visualización basada en bloques de 3D-AVC.

[0116] En 3D-HTM, la primera codificación de textura se aplica en condiciones de prueba comunes. Por lo tanto, la correspondiente visualización de profundidad no de base no está disponible al descodificar una visualización de textura no de base. Por lo tanto, la información de profundidad se estima y se utiliza para realizar la BVSP. Para estimar la información de profundidad para un bloque, se propone obtener primero un vector de disparidad a partir de los bloques contiguos, y luego usar el vector de disparidad obtenido para obtener un bloque de profundidad a partir de una visualización de referencia.

[0117] En el modelo de prueba HTM 5,1, existe un proceso para obtener un predictor de vector de disparidad, conocido como NBDV (vector de disparidad de bloque contiguo). (dv_x, dv_y) denota el vector de disparidad identificado por la función NBDV, y la posición del bloque actual es $(block_x, block_y)$. Se propuso buscar un bloque de profundidad en $(block_x + dv_x, block_y + dv_y)$ en la imagen en profundidad de la visualización de referencia. El bloque de profundidad capturado tendría el mismo tamaño que la unidad de predicción (PU) actual, y se usaría entonces para realizar una distorsión inversa para la PU actual. La FIG. 6 ilustra los pasos para ubicar un bloque de profundidad desde la visualización de referencia y luego usar el bloque de profundidad para la predicción de BVSP.

[0118] En el ejemplo de la FIG. 6, la imagen de profundidad 150 y la imagen de textura 154 corresponden a la misma visualización, mientras que la imagen de textura 152 corresponde a una visualización diferente. En particular, la imagen de textura 152 incluye el bloque actual 160 que está codificado en relación con la imagen de textura 154, actuando como imagen de referencia. Un codificador de vídeo puede referirse al bloque contiguo 162, que es contiguo al bloque actual 160. El bloque contiguo 162 incluye un vector de disparidad determinado previamente 166. El vector de disparidad 166 puede obtenerse como un vector de disparidad 164 para el bloque actual 160. Por lo tanto, el vector de disparidad 164 se refiere al bloque de profundidad 156 en la imagen de profundidad 150 de la visualización de referencia.

[0119] A continuación, el codificador de vídeo puede usar píxeles (es decir, valores de profundidad) del bloque de profundidad 156 para determinar los valores de disparidad 168 para píxeles (es decir, valores de textura) del bloque actual 160, para realizar distorsión inversa. A continuación, el codificador de vídeo puede sintetizar valores para un bloque predicho (es decir, un bloque de referencia BVSP) para el bloque actual 160 a partir de los píxeles identificados por los valores de disparidad 168. A continuación, el codificador de vídeo puede predecir el bloque actual 160 usando este bloque predicho. Por ejemplo, durante la codificación de vídeo mediante el codificador de vídeo 20, el codificador de vídeo 20 puede calcular las diferencias píxel por píxel entre el bloque predicho y el bloque actual 160 para producir un valor residual, que el codificador de vídeo 20 puede a continuación transformar, cuantificar y codificar por entropía. Por otro lado, durante la descodificación de vídeo mediante el descodificador de vídeo 30, el descodificador de vídeo 30 puede descodificar por entropía, cuantificar inversamente y transformar inversamente los datos residuales, a continuación combinar los datos residuales (en base de píxel por píxel) con el bloque predicho para reproducir el bloque actual 160.

[0120] JCT3V-C0152 propuso cambios a las técnicas BVSP de 3D-HEVC, como se describe a continuación. En particular, el texto en *cursiva* representa el texto agregado a 3D-HEVC, mientras que el texto entre corchetes precedido por "eliminado:" representa eliminaciones de 3D-HEVC:

Si BVSP está habilitado en la secuencia, el proceso NBDV para la predicción de movimiento entre visualizaciones se cambia y las diferencias se resaltan en los siguientes párrafos:

- Para cada uno de los bloques contiguos temporales, si utiliza un vector de movimiento de disparidad, el vector de movimiento de disparidad se devuelve como el vector de disparidad y se refina aún más con el procedimiento descrito en la Sección 1.6.1.3 de 3D-HEVC.
- *Para cada uno de los bloques contiguos espaciales, se aplica lo siguiente:*
 - *Para la lista de imágenes de referencia 0 y la lista de imágenes de referencia 1 en orden:*
 - Si utiliza un vector de movimiento de disparidad, el vector de movimiento de disparidad se devuelve como el vector de disparidad y se refina aún más con el procedimiento descrito en la Sección 1.6.1.3.

5 *▪ De lo contrario, si utiliza modo BVSP, el vector de movimiento asociado se devuelve como el vector de disparidad. Se refina aún más de una manera similar a la descrita en la Sección 1.6.1.3. Sin embargo, el valor de profundidad máximo se selecciona a partir de todos los píxeles del bloque de profundidad correspondiente en lugar de los cuatro píxeles de la esquina y el componente vertical del vector de disparidad refinado se establece en 0.*

• Para cada uno de los bloques contiguos espaciales, si usa un IDV, el IDV se devuelve como el vector de disparidad y se refina más con el procedimiento descrito en la Sección 1.6.1,3.

10 **[0121]** El modo de BVSP introducido se trata como un modo inter-codificado especial y se puede mantener un indicador indicando el uso del modo de BVSP para cada PU. En lugar de señalar el indicador en el flujo de bits, se agregó un nuevo candidato de fusión (candidato de fusión BVSP) a la lista de candidatos de fusión. El indicador depende de si el índice de candidato de fusión descodificado corresponde a un candidato de fusión BVSP. El candidato de fusión de BVSP se define, por JCT3V-C0152, de la forma siguiente:

- 15
- Índice de imágenes de referencia para cada lista de imágenes de referencia: -1
 - Vector de movimiento para cada lista de imágenes de referencia: el vector de disparidad refinado

20 **[0122]** En JCT3V-C0152, la posición insertada del candidato de fusión de BVSP depende de los bloques contiguos espaciales, como se analiza a continuación:

25 • Si cualquiera de los cinco bloques contiguos espaciales (A0, A1, B0, B1 o B2 mostrado en la FIG. 5) está codificado con el modo de BVSP, es decir, el indicador mantenido del bloque contiguo es igual a 1, el candidato de fusión de BVSP es tratado como el correspondiente candidato de fusión espacial e insertado en la lista de candidatos de fusión. El candidato de fusión de BVSP solo se insertará una vez en la lista de candidatos de fusión.

30 • De lo contrario (ninguno de los cinco bloques contiguos espaciales está codificado con el modo de BVSP), el candidato de fusión de BVSP se inserta en la lista de candidatos de fusión justo antes de los candidatos de fusión temporal.

[0123] Se observa que durante el proceso combinado de obtención del candidato de fusión bi-predictiva, se pueden comprobar condiciones adicionales para evitar incluir al candidato de fusión BVSP.

35 **[0124]** JCT3V-J0152 dispuso además que cada PU codificada por BVSP, con su tamaño indicado por NxM, se divide adicionalmente en varias sub-regiones con un tamaño igual a KxK (en el que K puede ser 4 o 2). Para cada sub-región, se obtiene un vector de movimiento de disparidad independiente y cada sub-región se predice a partir de un bloque ubicado mediante el vector de movimiento de disparidad obtenido en la imagen de referencia entre visualizaciones. En otras palabras, el tamaño de la unidad de compensación de movimiento para las PU codificadas mediante BVSP se fija en KxK píxeles. En condiciones de prueba comunes, K se fija en 4.

40 **[0125]** JCT3V-J0152 dispone además que para cada sub-región (bloque de 4x4 píxeles) dentro de una PU codificada con modo de BVSP, se ubica en primer lugar un correspondiente bloque de profundidad de 4x4 píxeles en la visualización de profundidad de referencia con el vector de disparidad refinado mencionado anteriormente. En segundo lugar, se selecciona el valor máximo de los dieciséis píxeles de profundidad en el correspondiente bloque de profundidad. En tercer lugar, el valor máximo se convierte en el componente horizontal de un vector de movimiento de disparidad. El componente vertical del vector de movimiento de disparidad se fija en 0.

50 **[0126]** En 3D-HEVC, cuando se aplica el orden de codificación de textura primero, para cada unidad de predicción (PU), un vector de disparidad se podría obtener de NBDV con/sin considerar los valores de profundidad en la visualización de profundidad de referencia. Después de obtener un vector de disparidad, se refinaría aún más para cada sub-región 4x4 de una PU, si la PU se codifica con el modo BVSP.

55 **[0127]** 3D-HEVC describe el proceso de refinamiento que incluye dos pasos: 1) seleccionar un valor de profundidad máxima del bloque de profundidad 4x4 en la visualización de profundidad de referencia que se ubica mediante el vector de disparidad obtenido; 2) convertir el valor de profundidad en un componente horizontal del vector de disparidad refinado mientras se mantiene el componente vertical del vector de disparidad refinado para que sea 0. Después de que el vector de disparidad se refine para una sub-región 4x4 de una PU, 3D-HEVC dispone que el vector de disparidad refinado se usa para ubicar un bloque en la visualización de textura de referencia para la compensación de movimiento.

60 **[0128]** A continuación se analizará NBDV en 3D-AVC. De acuerdo con algunas técnicas de codificación 3D, el NBDV a nivel de MB se puede usar para obtener un vector de disparidad para el MB actual y se usa además para la predicción de vectores de movimiento. Una vez que se identifica un vector de movimiento de disparidad, es decir, uno de los bloques contiguos temporales o espaciales usa la imagen de referencia entre visualizaciones, se devuelve como el vector de disparidad para el MB actual.

[0129] De acuerdo con algunas técnicas de codificación 3D, el vector de disparidad obtenido de NBDV se puede refinar aún más mediante el acceso al bloque de profundidad correspondiente. Por ejemplo, los valores de profundidad de las cuatro esquinas del bloque de profundidad de la visualización de referencia (identificados mediante el vector de disparidad) se pueden usar y el valor de profundidad máximo se elige y convierte en un vector de disparidad. De acuerdo con otras técnicas de codificación 3D, BVSP se utiliza de una manera para cada bloque 4x4 u 8x8, el bloque de profundidad correspondiente se usa para producir el vector de movimiento de disparidad. De manera similar al proceso de obtención de vector de disparidad, las cuatro esquinas del bloque de profundidad (identificadas mediante el vector de disparidad refinado) se pueden usar y el valor de profundidad máximo se elige para convertirlo en un vector de movimiento de disparidad.

[0130] Las técnicas de esta divulgación pueden ocuparse de varios posibles problemas. Como ejemplo, para D-MVP en modo inter, la conversión de profundidad a DV debe realizarse varias veces para obtener DV para varios bloques de partición en un MB, lo cual aumenta el número de muestras de profundidad a las que se puede acceder desde la memoria cuando el MB actual se divide en particiones/sub-bloques. Como otro ejemplo, para D-MVP en modo de salto/directo, se obtiene un vector de disparidad para un MB al tomar el valor de profundidad máximo de cuatro muestras de esquina del bloque de profundidad asociado; esto requiere un alto ancho de banda de acceso a la memoria. Como otro ejemplo, en el modo BVSP, se obtiene un vector de disparidad para una sub-región de tamaño K x K tomando el máximo de cuatro muestras de profundidad de esquina del bloque de profundidad asociado de la sub-región actual; esto se hace a costa del alto ancho de banda de acceso a la memoria. Como otro ejemplo, para los MB de salto/directos, cuando el vector de movimiento entre visualizaciones no está disponible, se utiliza la predicción del vector de movimiento basado en la mediana convencional. Como otro ejemplo, en 3D-HEVC, refinar el vector de disparidad para un bloque actual cuando el vector de disparidad se obtiene de un bloque contiguo predicho entre visualizaciones solo requiere acceder a las cuatro esquinas del bloque de profundidad de referencia. Refinar el vector de disparidad para un bloque actual cuando el vector de disparidad se obtiene de un bloque contiguo codificado con BVSP y producir un vector de movimiento de disparidad para una sub-región 4x4 u 8x8 para una PU codificada con modo BVSP requiere un diseño ligeramente diferente, que requiere acceso a todas las muestras del bloque de referencia.

[0131] Esta divulgación introduce técnicas que pueden simplificar el proceso de obtención de vectores de disparidad (movimiento), cuando los vectores de disparidad se obtienen al acceder a un mapa de profundidad. La explicación de estas técnicas puede centrarse principalmente en 3D-AVC, aunque debe entenderse que ideas similares pueden ser aplicables a 3D-HEVC. Más específicamente, esta divulgación proporciona un esquema de obtención de vectores de disparidad simplificado y unificado para varios modos de predicción inter (incluyendo salto/directo, inter) así como modos BVSP, cuando se utiliza un esquema de conversión de profundidad a disparidad. Los cálculos y accesos de memoria pueden simplificarse.

[0132] La FIG. 7 muestra un ejemplo de un bloque de profundidad 8x8 170. Las cuatro muestras de esquina del bloque de profundidad 170 están etiquetadas como 172A-172D. Un "reduced_resolution_flag" en el flujo de bits puede indicar si una visualización de profundidad tiene una resolución reducida en relación con la visualización de textura o si la visualización de profundidad tiene la misma resolución que la visualización de textura. Si reduced_resolution_flag es igual a 1, entonces un macrobloque 16x16 en una visualización de textura tendrá un bloque de profundidad 8x8 correspondiente (es decir, co-ubicado), como el bloque de profundidad 8x8 170, en una visualización de textura correspondiente. Si reduced_resolution_flag es igual a 0, entonces un macrobloque 16x16 en una visualización de textura tendrá un bloque de profundidad de 16x16 correspondiente (es decir, co-ubicado) en una visualización de textura correspondiente. Varios ejemplos a continuación se explicarán haciendo referencia al bloque de profundidad 170.

[0133] De acuerdo con una técnica de esta divulgación, cuando se codifica un componente de visualización de textura no básica, el descodificador de vídeo 30 puede obtener solo un vector de disparidad para un macrobloque completo al acceder a un bloque de profundidad correspondiente al macrobloque que contiene el bloque de textura actual. Independientemente de si el MB actual está codificado con modos de salto/directo u otros modos inter, siempre que se requiera un vector de disparidad para cualquier bloque dentro del MB, el descodificador de vídeo 30 puede obtener el mismo vector de disparidad único una vez de la misma obtención de vectores de disparidad de nivel MB. El descodificador de vídeo 30 puede, por ejemplo, obtener el vector de disparidad accediendo a los valores de profundidad de las muestras de esquina 172A-172D y determinando un valor de profundidad máximo de las cuatro muestras de esquina. El descodificador de vídeo 30 puede convertir el valor máximo de profundidad en un valor de disparidad utilizando, por ejemplo, una tabla de consulta o las ecuaciones (1) y (2) anteriores. En algunas implementaciones, el descodificador de vídeo 30 puede obtener un vector de disparidad al acceder a valores de profundidad diferentes a las cuatro muestras de esquina y/o al identificar un valor diferente al máximo.

[0134] Como ejemplo, para un macrobloque de una visualización de textura no básica, el descodificador de vídeo 30 puede ubicar un bloque de profundidad que corresponde al macrobloque, y basándose en al menos un valor de profundidad del bloque de profundidad, obtener un vector de disparidad para el macrobloque. El descodificador de vídeo 30 puede codificar un primer sub-bloque del macrobloque basándose en el vector de disparidad obtenido y codificar un segundo sub-bloque del macrobloque basándose en el vector de disparidad obtenido. El bloque de

profundidad y el macrobloque pueden estar co-ubicados. El descodificador de vídeo 30 puede obtener el vector de disparidad para el macrobloque al determinar un conjunto de valores de profundidad que incluye valores de profundidad de dos o más muestras de esquina del bloque de profundidad, e identificar a partir del conjunto de valores de profundidad, un valor de profundidad máxima. El descodificador de vídeo 30 puede, por ejemplo, codificar el primer sub-bloque usando uno de un modo de salto y un modo directo, y codificar el segundo sub-bloque usando un modo de predicción inter diferente al modo de salto o el modo directo.

[0135] Una posible diferencia entre los modos de salto/directo y la predicción inter es que cuando un macrobloque se codifica utilizando los modos de salto/directo, no habrá ninguna partición, es decir, el tamaño de MB de 16x16 se codifica completamente. Cuando un MB se codifica utilizando un modo que no sea de salto/directo (es decir, modos inter), puede haber una partición de MB. Además, cada partición de MB puede dividirse en sub-bloques (es decir, sub-bloques H.264/AVC). De acuerdo con las técnicas de esta divulgación, el mismo vector de disparidad obtenido del MB se puede usar para todos los modos, tales como modos de salto/directo y modos inter.

[0136] De acuerdo con otra técnica de esta divulgación, cuando se obtiene un vector de disparidad de un bloque de profundidad correspondiente del componente de visualización de profundidad dentro de la misma visualización, solo se accede a las muestras de profundidad de esquina inferior izquierda (172A) e inferior derecha (172B) del bloque de profundidad correspondiente del MB actual. Como se muestra en la FIG. 7, típicamente un MB corresponde a un bloque de referencia 8x8 en la misma visualización en 3D-AVC. El bloque de profundidad típicamente tiene una resolución espacial más baja (con el muestreo de horizontal y vertical reducido a la mitad). Por lo tanto, un macrobloque 16x16 corresponde a un bloque de profundidad 8x8. De acuerdo con una técnica de esta divulgación, solo se accede a las muestras de esquina 172A y 172B para obtener el vector de disparidad para todo el MB. Además, el valor de profundidad máxima de las muestras de profundidad a las que se accede se utiliza para convertirlo en un vector de disparidad. De forma alternativa, solo se accede a las muestras de esquina superior izquierda (172C) e inferior derecha (172B) del bloque de profundidad de referencia. De forma alternativa, solo se accede a las muestras de esquina superior izquierda (172C) y superior derecha (172D) del bloque de profundidad de referencia. De forma alternativa, solo se accede a las muestras de esquina inferior izquierda (172A) e inferior derecha (172B) del bloque de profundidad de referencia. De forma alternativa, solo se accede a las muestras de esquina inferior izquierda (172A) y superior derecha (172D) del bloque de profundidad de referencia. De forma alternativa, solo se accede a las muestras de esquina inferior derecha (172B) y superior derecha (172D) del bloque de profundidad de referencia. De forma alternativa, se accede a otras dos muestras ubicadas dentro del bloque de profundidad correspondiente, por ejemplo, un píxel central y uno de esquina o dos píxeles centrales, etc. De forma alternativa, cuando se accede a dos muestras, puede utilizarse el promedio/mínimo de estas dos muestras de profundidad, en lugar de máximo.

[0137] De acuerdo con otra técnica de esta divulgación, se puede usar una metodología de conversión de profundidad a disparidad similar en los modos BVSP para obtener un vector de disparidad para cada sub-región de tamaño $K \times K$ dentro de una partición de MB. Primero se identifica un bloque de profundidad de referencia para una sub-región $K \times K$, y se accede a las mismas muestras de esquina con posiciones en una coordinación relativa (por ejemplo, las esquinas inferior derecha e inferior izquierda del bloque de profundidad de referencia correspondiente/co-ubicado a la sub-región $K \times K$) del bloque de profundidad de referencia, similar a los de la obtención de vectores de disparidad de nivel MB. De forma alternativa, se puede acceder a menos muestras de esquina de un bloque de profundidad de referencia para BVSP.

[0138] De acuerdo con otra técnica de esta divulgación, en los modos de salto/directo de la 3D-AVC actual, cuando el vector de movimiento entre visualizaciones no está disponible, el predictor del vector de movimiento se establece como el vector de movimiento del primer contiguo espacial disponible, incluyendo el índice de referencia, en lugar de utilizar el medio de vectores de movimiento de tres bloques contiguos. Compruebe en orden, si alguno de los contiguos espaciales tiene un índice de imagen de referencia mayor o igual a 0. Si es verdadero, el vector de movimiento y el índice de imagen de referencia del MB actual se establecen respectivamente para que sean iguales al vector de movimiento y el índice de referencia del bloque contiguo espacial, y el proceso de comprobación finaliza.

[0139] De acuerdo con otra técnica de esta divulgación, cuando se usa NBDV para 3D-AVC y el mapa de profundidad de la visualización de referencia se usa para refinar el vector de disparidad, las mismas dos muestras de esquina se pueden usar para obtener un valor de profundidad óptimo para convertir en un vector de disparidad refinado.

[0140] De acuerdo con otra técnica de esta divulgación, cuando se usa NBDV para 3D-AVC y el mapa de profundidad de la visualización de referencia se usa para producir vectores de movimiento de disparidad en el modo BVSP, se pueden usar las mismas dos muestras de esquina para obtener un valor de profundidad óptimo para convertirse en un vector de movimiento de disparidad.

[0141] De acuerdo con otra técnica de esta divulgación, en 3D-HEVC, de manera similar, el acceso al bloque de referencia para un vector de movimiento de disparidad en el modo BVSP puede alinearse de una manera simplemente comprobando las muestras de esquina de los bloques contiguos. En otras palabras, para cada sub-región $K \times K$, el descodificador de vídeo 30 comprueba solo las muestras de esquina del bloque de profundidad de referencia de la sub-región $K \times K$ identificada por un vector de disparidad de la unidad de predicción que contiene la sub-región $K \times K$.

Además, las simplificaciones descritas anteriormente pueden aplicarse a 3D-HEVC, aplicando refinamiento a una PU o CU en lugar de un bloque de MB.

5 **[0142]** Los aspectos de la implementación de las técnicas descritas anteriormente se explicarán ahora con mayor detalle, comenzando con la obtención de vectores de disparidad en el modo de salto/directo. Para este ejemplo, la ubicación de la muestra superior izquierda del MB actual en relación con la imagen actual se denota con (x, y). Se selecciona un valor de profundidad (D) para los MB actuales de los píxeles de las esquinas inferior izquierda e inferior derecha en la visualización de profundidad de referencia como

10
$$D = \text{máx}(D_0, D_1),$$

donde la función $\text{máx}(\bullet)$ devuelve el valor máximo de D_i (siendo i un valor de 0 a 1) y D_i indica el valor de profundidad en la i -ésima ubicación de píxel que se calcula como

15
$$i=0: (x \gg \text{reduced_resolution_flag}, (y \gg \text{reduced_resolution_flag}) + \text{SMD_POS}),$$

$$i=1: ((x \gg \text{reduced_resolution_flag}) + \text{SMD_POS}, (y \gg \text{reduced_resolution_flag}) + \text{SMD_POS}),$$

20 En este caso, SMD_POS es igual a 7 y 15 cuando la "reduced_resolution_flag", respectivamente, es igual a 1 y 0. "reduced_resolution_flag" igual a 1 especifica que los componentes de visualización de profundidad de un par de componentes de visualización tienen una resolución espacial más baja que el componente de luma del componente de visualización de textura del mismo par de componentes de visualización, y la anchura y la altura de los componentes de visualización de profundidad tienen ambos la mitad de la anchura y la altura de todos los componentes de visualización de textura. "reduced_resolution_flag" igual a 0 especifica que cuando están presentes tanto los componentes de visualización de profundidad como los componentes de visualización de textura, tienen la misma resolución espacial. Finalmente, el componente horizontal del vector de disparidad se calcula a partir del valor de profundidad seleccionado D utilizando la ecuación. (1) y (2) y el componente vertical del vector de disparidad se establece siempre en 0.

30 **[0143]** Ahora se describirá la obtención de vectores de disparidad en el modo inter. En este ejemplo, la posición superior izquierda de los MB actuales (donde se ubica la partición/sub-bloque actual) con respecto a la imagen actual se denota con (x, y). Se selecciona un valor de profundidad (D) para los MB actuales de los píxeles de las esquinas inferior izquierda e inferior derecha en la visualización de profundidad de referencia como

35
$$D = \text{máx}(D_0, D_1),$$

donde la función $\text{máx}(\bullet)$ devuelve el valor máximo de D_i (siendo i un valor de 0 a 1) y D_i indica el valor de profundidad en la i -ésima ubicación de píxel que se calcula como

40
$$i=0: (x \gg \text{reduced_resolution_flag}, (y \gg \text{reduced_resolution_flag}) + \text{SMD_POS}),$$

$$i=1: ((x \gg \text{reduced_resolution_flag}) + \text{SMD_POS}, (y \gg \text{reduced_resolution_flag}) + \text{SMD_POS}),$$

45 aquí, SMD_POS y reduced_resolution_flag funcionan como se describió anteriormente. Finalmente, el componente horizontal del vector de disparidad se calcula a partir del valor de profundidad seleccionado D utilizando la ecuación. (1) y (2), o una tabla de consulta, y el componente vertical del vector de disparidad se puede establecer en 0.

50 **[0144]** Ahora se describirá la obtención de vectores de disparidad en el modo BVSP. El tamaño de la sub-región por $K \times K$ (donde K podría ser 8 o 4), la posición superior izquierda de una sub-región dentro de la partición de MB actual relativa a la imagen actual se denota con (x, y). Se selecciona un valor de profundidad (D) para cada sub-región de tamaño $K \times K$ desde los píxeles de las esquinas inferior izquierda e inferior derecha en la visualización de profundidad de referencia como

55
$$D = \text{máx}(D_0, D_1),$$

donde la función $\text{máx}(\bullet)$ devuelve el valor máximo de D_i (siendo i de 0 a 1) y D_i indica la ubicación del píxel de profundidad i -ésimo en:

60
$$i=0: ((x \gg \text{reduced_resolution_flag}), (y \gg \text{reduced_resolution_flag}) + \text{VSP_S})$$

$$i=1: ((x \gg \text{reduced_resolution_flag}) + \text{VSP_S}, (y \gg \text{reduced_resolution_flag}) + \text{VSP_S})$$

aquí, VSP_S se calcula como:

65
$$\text{VSP_S} = (K \gg \text{reduced_resolution_flag}) - 1,$$

y las funciones `reduced_resolution_flag` como se definieron anteriormente. Finalmente, el componente horizontal del vector de disparidad puede calcularse a partir del valor de profundidad D seleccionado utilizando la ecuación (1) y (2), o las tablas de consulta, y el componente vertical del vector de disparidad se puede establecer en 0.

[0145] Ahora se analizará el MVP en los modos de salto/directo cuando el movimiento entre visualizaciones no está disponible. En este ejemplo, los vectores de movimiento $Mvc_X[i]$ y los índices de referencia $Ric_X[i]$ (para X igual a 0 y 1) con i igual a 0, 1 y 2 se indican con respecto a los contiguos espaciales A, B y C del MB actual. Cuando el bloque de referencia en la visualización de base correspondiente al MB de salto/directo actual se codifica en el modo intra, el vector de movimiento predicho ($Mvp0$, $Mvp1$) y el índice de referencia ($Ri0$, $Ri1$) para la MB actual se calcula de la siguiente manera:

1. inicializar $Mvp0$, $Mvp1$ igual al vector de movimiento cero $[0,0]$ y el índice de referencia $Ri0$, $Ri1$ se establece igual a -1.

2. Cuando todos los contiguos candidatos espaciales tienen el índice de referencia igual a -1 (es decir, $Ric_0[i]=-1$ y $Ric_1[i]=-1$, para todos $i = 0, 1, 2$), entonces el vector de movimiento y el índice de referencia del MB actual se establece en cero.

3. Para i de 0 a 2, se aplica lo siguiente:

- Si el contiguo espacial i -ésimo ($i = 0, 1, 2$) tiene un índice de referencia mayor o igual a cero, configure el vector de movimiento predicho $Mvp0$ y $Mvp1$ para que sea igual al vector de movimiento $Mvc_0[i]$ y $Mvc_1[i]$, respectivamente. Además, el índice de referencia $Ri0$ y $Ri1$ se establece igual al índice de referencia $Ric_0[i]$ y $Ric_1[i]$, respectivamente.

[0146] La FIG. 8 es un diagrama de bloques que ilustra un codificador de vídeo de ejemplo que puede implementar las técnicas descritas en esta divulgación. Por ejemplo, la FIG. 8 ilustra el codificador de vídeo 20 que puede representar un codificador de vídeo conforme a la 3D-AVC o a la 3D-HEVC. El codificador de vídeo 20 se describirá usando cierta terminología de la HEVC, tal como PU, TU y CU, pero debería entenderse que las técnicas descritas con referencia al codificador de vídeo 20 también pueden realizarse con vídeo codificado de acuerdo con la norma H.264.

[0147] El codificador de vídeo 20 puede realizar la intracodificación y la intercodificación de bloques de vídeo dentro de fragmentos de vídeo. Por ejemplo, el codificador de vídeo 20 puede realizar la codificación de predicción inter o la codificación de predicción intra. La intracodificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial en el vídeo dentro de una trama o imagen de vídeo dada. La inter-codificación se basa en la predicción temporal o en la predicción entre visualizaciones para reducir o eliminar la redundancia temporal dentro de tramas o imágenes contiguas de una secuencia de vídeo o la redundancia entre imágenes en diferentes visualizaciones. El modo intra (modo I) puede referirse a cualquiera de varios modos de compresión espacial. Los modos inter, tales como la predicción unidireccional (modo P) o la predicción bidireccional (modo B), pueden referirse a cualquiera de varios modos de compresión temporal.

[0148] En el ejemplo de la FIG. 8, el codificador de vídeo 20 incluye una memoria de datos de vídeo 40, una unidad de procesamiento de predicciones 42, una memoria de imágenes de referencia 64, un sumador 50, una unidad de procesamiento de transformadas 52, una unidad de procesamiento de cuantificación 54 y una unidad de codificación por entropía 56. La unidad de procesamiento de predicción 42 incluye la unidad de estimación de movimiento y disparidad 44, la unidad de compensación de movimiento y disparidad 46 y la unidad de predicción intra 48. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 también incluye una unidad de procesamiento de cuantificación inversa 58, una unidad de procesamiento de transformadas inversas 60 y un sumador 62. También se puede incluir un filtro de eliminación de bloques (no mostrado en la FIG. 8) para filtrar fronteras de bloques, para eliminar distorsiones de efecto pixelado del vídeo reconstruido. Si se desea, el filtro de eliminación de bloques filtrará habitualmente la salida del sumador 62. También se pueden usar filtros de bucle adicionales (en el bucle o tras el bucle), además del filtro de eliminación de bloques.

[0149] La memoria de datos de vídeo 40 puede almacenar datos de vídeo a codificar por parte de los componentes del codificador de vídeo 20. Los datos de vídeo almacenados en la memoria de datos de vídeo 40 se pueden obtener, por ejemplo, a partir del origen de vídeo 18. La memoria de imágenes de referencia 64 es un ejemplo de una memoria intermedia de imágenes de descodificación (DPB), que almacena datos de vídeo de referencia para su uso en la codificación de datos de vídeo por el codificador de vídeo 20 (por ejemplo, en los modos de intra-codificación o inter-codificación, también denominados modos de codificación de predicción intra o predicción inter). La memoria de datos de vídeo 40 y la memoria de imágenes de referencia 64 pueden ser formados por cualquiera entre varios dispositivos de memoria, tales como memoria dinámica de acceso aleatorio (DRAM), incluyendo DRAM síncrona (SDRAM), RAM magnetorresistiva (MRAM), RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. La memoria de datos de vídeo 40 y la memoria de imágenes de referencia 64 pueden ser proporcionadas por el mismo dispositivo de memoria

o por dispositivos de memoria independientes. En diversos ejemplos, la memoria de datos de vídeo 40 puede estar en un chip con otros componentes del codificador de vídeo 20, o fuera de chip con respecto a esos componentes.

[0150] El codificador de vídeo 20 recibe datos de vídeo y una unidad de partición (no mostrada) divide los datos en bloques de vídeo. Esta partición también puede incluir la partición en fragmentos, mosaicos u otras unidades más grandes, así como la partición de bloques de vídeo (por ejemplo, particiones de macrobloques y sub-bloques de particiones). El codificador de vídeo 20 ilustra, en general, los componentes que codifican bloques de vídeo dentro de un fragmento de vídeo que se va a codificar. El fragmento se puede dividir en múltiples bloques de vídeo (y, posiblemente, en conjuntos de bloques de vídeo denominados mosaicos). La unidad de procesamiento de predicción 42 puede seleccionar una entre una pluralidad de posibles modos de codificación, tal como una entre una pluralidad de modos de intra-codificación (modos de codificación de predicción intra) o una entre una pluralidad de modos de inter-codificación (modos de codificación de predicción inter), para el bloque de vídeo actual, basándose en los resultados de errores (por ejemplo, la velocidad de codificación y el nivel de distorsión). La unidad de procesamiento de predicción 42 puede, por ejemplo, seleccionar un modo BVSP para codificar un bloque actual. La unidad de procesamiento de predicción 42 puede proporcionar el bloque intracodificado 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 imagen de referencia.

[0151] La unidad de predicción intra 48, dentro de la unidad de procesamiento de predicción 42, puede realizar la codificación de predicción intra del bloque de vídeo actual con respecto a uno o más bloques contiguos en la misma trama o fragmento que el bloque actual a codificar, para proporcionar compresión espacial. La unidad de estimación de movimiento y disparidad 44 y la unidad de compensación de movimiento y disparidad 46 dentro de la unidad de procesamiento de predicción 42 llevan a cabo la codificación de predicción inter del bloque de vídeo actual en relación con uno o más bloques predictivos en una o más imágenes de referencia para proporcionar compresión temporal.

[0152] La unidad de estimación de movimiento y disparidad 44 puede estar configurada para determinar el modo de predicción inter para un fragmento de vídeo de acuerdo con un patrón predeterminado para una secuencia de vídeo. El patrón predeterminado puede designar fragmentos de vídeo en la secuencia como fragmentos P o fragmentos B. La unidad de estimación de movimiento y disparidad 44 y la unidad de compensación de movimiento y disparidad 46 pueden estar sumamente integradas, pero se ilustran por separado con fines conceptuales. La estimación de movimiento, llevada a cabo por la unidad de estimación de movimiento y disparidad 44, es el proceso de generar vectores de movimiento, que estiman el movimiento para los bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de un bloque de vídeo dentro de una trama o imagen de vídeo actual a un bloque predictivo dentro de una imagen de referencia.

[0153] Un bloque predictivo es un bloque que se revela como estrechamente coincidente con el bloque de vídeo a codificar, en lo que respecta a la diferencia de píxeles, lo cual puede determinarse mediante suma de diferencias absolutas (SAD), suma de las diferencias al cuadrado (SSD) u otras métricas de diferencia. En algunos ejemplos, el codificador de vídeo 20 puede calcular valores para posiciones fraccionarias de píxeles de imágenes de referencia almacenadas en la memoria de imágenes de referencia 64. Por ejemplo, el codificador de vídeo 20 puede interpolar valores de posiciones de un cuarto de píxel, posiciones de un octavo de píxel u otras posiciones fraccionarias de píxel de la imagen de referencia. Por lo tanto, la unidad de estimación de movimiento y disparidad 44 puede realizar una búsqueda de movimiento en relación con las posiciones de píxeles completas y las posiciones de píxeles fraccionarias, y emitir un vector de movimiento con una precisión de píxel fraccionaria.

[0154] La unidad de estimación de movimiento y disparidad 44 calcula un vector de movimiento para un bloque de vídeo en un fragmento inter-codificado (codificado por predicción inter) comparando la posición del bloque de vídeo con la posición de un bloque predictivo de una imagen de referencia. La imagen de referencia se puede seleccionar entre una primera lista de imágenes de referencia (RefPicList0) o una segunda lista de imágenes de referencia (RefPicList1), cada una de las cuales identifica una o más imágenes de referencia almacenadas en la memoria de imágenes de referencia 64. La unidad de estimación de movimiento y disparidad 44 envía el vector de movimiento calculado a la unidad de codificación por entropía 56 y a la unidad de compensación de movimiento y disparidad 46.

[0155] La compensación de movimiento, realizada por la unidad de compensación de movimiento y disparidad 46, puede implicar extraer o generar el bloque predictivo basándose en el vector de movimiento determinado mediante estimación de movimiento, realizando posiblemente interpolaciones hasta una precisión de sub-píxel. Tras recibir el vector de movimiento para el bloque de vídeo actual, la unidad de compensación de movimiento y disparidad 46 puede ubicar el bloque predictivo al que apunta el vector de movimiento en una de las listas de imágenes de referencia. El codificador de vídeo 20 forma un bloque de vídeo residual restando los valores de píxel del bloque predictivo a los valores de píxel del bloque de vídeo actual que se está codificando, formando valores de diferencia de píxel. Los valores de diferencia de píxel forman datos residuales para el bloque, y pueden incluir componentes de diferencia tanto de luma como de croma. El sumador 50 representa el componente o los componentes que realizan esta operación de resta. La unidad de compensación de movimiento y disparidad 46 también puede generar elementos sintácticos asociados a los bloques de vídeo y al fragmento de vídeo para su uso por el descodificador de vídeo 30 en la descodificación de los bloques de vídeo del fragmento de vídeo.

5 **[0156]** En algunos ejemplos, la unidad de compensación de movimiento y disparidad 46 puede realizar BVSP para un bloque actual en una visualización actual. Es decir, la unidad de compensación de movimiento y disparidad 46 puede determinar una imagen de la memoria de imágenes de referencia 64 en una primera visualización. Como se explicó con mayor detalle anteriormente, la unidad de compensación de movimiento y disparidad 46 puede determinar un vector de disparidad para el bloque actual al acceder al bloque correspondiente en una visualización en profundidad. A continuación, al utilizar los valores de profundidad del bloque de profundidad, la unidad de compensación de movimiento y disparidad 46 puede distorsionar los valores de píxeles de una imagen en la primera visualización, determinada en relación con las posiciones de píxeles en el bloque actual, de manera que se forme un bloque predicho en una segunda visualización, diferente de la primera visualización y la visualización actual. La unidad de compensación de movimiento y disparidad 46 puede proporcionar este bloque previsto para el sumador 50 y el sumador 62, para su uso en el cálculo de un residuo y en la reproducción del bloque actual, respectivamente. Del mismo modo, de acuerdo con las técnicas de esta divulgación, el codificador de vídeo 20 puede codificar datos de sintaxis que definen información de movimiento para el bloque actual de tal manera que la información de movimiento incluya un índice de referencia que tenga un valor que identifique la imagen en la primera visualización desde la cual el bloque predicho (es decir, un bloque de referencia BVSP) se sintetiza.

20 **[0157]** La unidad de predicción intra 48 puede realizar predicción inter de un bloque actual, de forma alternativa a la predicción inter llevada a cabo por la unidad de estimación de movimiento y disparidad 44 y la unidad de compensación de movimiento y disparidad 46, como se ha descrito anteriormente. En particular, la unidad de predicción intra 48 puede determinar un modo de predicción intra a usar para codificar un bloque actual. En algunos ejemplos, la unidad de predicción intra 48 puede codificar un bloque actual usando diversos modos de predicción intra, por ejemplo, durante pasadas de codificación independientes, y la unidad de predicción intra 48 (o una unidad de selección de modo, en algunos ejemplos) puede seleccionar un modo adecuado de predicción intra a usar a partir de los modos probados. Por ejemplo, la unidad de predicción intra 48 puede calcular valores de velocidad-distorsión usando un análisis de velocidad-distorsión para los diversos modos de predicción intra probadas, y seleccionar el modo de predicción intra que tenga las mejores características de velocidad-distorsión entre los modos probados. El análisis de velocidad-distorsión determina, en general, una magnitud de distorsión (o error) entre un bloque codificado y un bloque original no codificado que fue codificado para generar el bloque codificado, así como una velocidad de bits (es decir, un número de bits) usada para generar el bloque codificado. La unidad de predicción intra 48 puede calcular proporciones a partir 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 **[0158]** En cualquier caso, tras seleccionar un modo de predicción intra para un bloque, la unidad de predicción intra 48 puede proporcionar información, que indica el modo de predicción intra seleccionada para el bloque, a la unidad de codificación por entropía 56. La unidad de codificación por entropía 56 puede codificar la información que indica el modo de predicción intra seleccionado de acuerdo con las técnicas de esta divulgación. El codificador de vídeo 20 puede incluir, en el flujo de bits transmitido, datos de configuración, que pueden incluir una pluralidad de tablas de índices de modo de predicción intra y una pluralidad de tablas modificadas de índices de modo de predicción intra (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 de modo de predicción intra y una tabla modificada de índices de modo de predicción intra, a usar para cada uno de los contextos.

45 **[0159]** Después de que la unidad de procesamiento de predicción 42 genera el bloque predictivo para el bloque de vídeo actual, ya sea mediante la predicción inter o la predicción intra, el codificador de vídeo 20 forma un bloque de vídeo residual restando el bloque predictivo al bloque de vídeo actual. Los datos de vídeo residual en el bloque residual se pueden aplicar a la unidad de procesamiento de transformadas 52. La unidad de procesamiento de transformada 52 transforma los datos de vídeo residuales en coeficientes de transformada residuales usando una transformada, tal como una transformada de coseno discreta (DCT) o una transformada conceptualmente similar. La unidad de procesamiento de transformada 52 puede convertir los datos de vídeo residuales de un dominio de píxel a un dominio de transformada, tal como un dominio de frecuencia.

55 **[0160]** La unidad de procesamiento de transformada 52 puede enviar los coeficientes de transformada resultantes a la unidad de procesamiento de cuantificación 54. La unidad de procesamiento de cuantificación 54 cuantifica los coeficientes de transformada para reducir adicionalmente la velocidad de bits. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos, o a la totalidad, de los coeficientes. El grado de cuantificación se puede modificar ajustando un parámetro de cuantificación. En algunos ejemplos, la unidad de procesamiento de cuantificación 54 luego puede realizar un barrido de la matriz, incluyendo los coeficientes de transformada cuantificados. De forma alternativa, la unidad de codificación por entropía 56 puede realizar la exploración.

60 **[0161]** Tras la cuantificación, la unidad de codificación por entropía 56 realiza la codificación por entropía de los coeficientes de transformada cuantificados. Por ejemplo, la unidad de codificación por entropía 56 puede realizar una codificación de longitud variable adaptativa al contexto (CAVLC), una codificación aritmética binaria adaptativa al contexto (CABAC), una codificación aritmética binaria adaptativa al contexto basada en la sintaxis (SBAC), una codificación por entropía por división de intervalos de probabilidad (PIPE) u otros procedimientos o técnicas de codificación por entropía. Tras la codificación por entropía por la unidad de codificación por entropía 56, el flujo de bits codificado se puede transmitir al descodificador de vídeo 30, o archivar para su posterior transmisión o recuperación

por el decodificador de vídeo 30. La unidad de codificación por entropía 56 también puede realizar la codificación por entropía de los vectores de movimiento y los otros elementos sintácticos para el fragmento de vídeo actual que se está codificando.

5 **[0162]** La unidad de procesamiento de cuantificación inversa 58 y la unidad de procesamiento de transformada inversa 60 aplican una cuantificación inversa y una transformación inversa, respectivamente, para reconstruir el bloque residual en el dominio de píxel, para su posterior uso como un bloque de referencia de una imagen de referencia. La unidad de compensación de movimiento y disparidad 46 puede calcular un bloque de referencia sumando el bloque residual a un bloque predictivo de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. La unidad de compensación de movimiento y disparidad 46 también puede aplicar uno o más filtros de interpolación al bloque residual reconstruido para calcular valores de píxeles fraccionarios, para su uso en la estimación de movimiento. El sumador 62 añade el bloque residual reconstruido al bloque de predicción compensado en movimiento, generado por la unidad de compensación de movimiento y disparidad 46 para generar un bloque de referencia para su almacenamiento en la memoria de imágenes de referencia 64. La unidad de estimación de movimiento y disparidad 44 y la unidad de compensación de movimiento y disparidad 46 pueden usar el bloque de referencia como un bloque de referencia para realizar predicción inter de un bloque en una posterior imagen o trama de vídeo.

20 **[0163]** De esta manera, el codificador de vídeo 20 es un ejemplo de un codificador de vídeo que puede configurarse para implementar una o más técnicas a modo de ejemplo descritas en esta divulgación. Por ejemplo, la memoria de datos de vídeo 40 almacena datos de vídeo. Los datos de vídeo pueden incluir un componente de vídeo de textura de una visualización dependiente y un componente de visualización de profundidad que corresponde al componente de visualización de textura, cada uno de los cuales ha de codificar el codificador de vídeo 20 en un proceso de codificación de vídeo conforme a la norma 3D-AVC o la norma 3D-HEVC.

25 **[0164]** En las técnicas descritas en esta divulgación, el codificador de vídeo 20 puede incluir uno o más procesadores que están configurados para codificar, en un proceso de codificación de vídeo conforme a 3D-AVC o a 3D-HEVC, un componente de visualización de textura de una visualización dependiente de los datos de vídeo. Como se ha descrito anteriormente, cada visualización en una norma 3D-AVC incluye un componente de visualización de textura y un componente de visualización de profundidad. Existe una visualización básica y una o más visualizaciones de realce o dependientes en la norma 3D-AVC, donde los componentes de visualización de textura de las una o más visualizaciones de realce o dependientes se pueden predecir entre visualizaciones.

35 **[0165]** Para codificar el componente de visualización de textura, el codificador de vídeo 20 se puede configurar para evaluar la información de movimiento de uno o más bloques contiguos de un bloque actual en el componente de visualización de textura, para determinar si al menos un bloque contiguo se predice entre visualizaciones con un vector de movimiento de disparidad que se refiere a una imagen de referencia entre visualizaciones en una visualización distinta a la visualización dependiente. El codificador de vídeo 20 puede obtener un vector de disparidad para el bloque actual basándose en el vector de movimiento de disparidad para uno de los bloques contiguos. Para la codificación de textura en primer lugar, el codificador de vídeo 20 puede codificar un componente de visualización de profundidad, de los datos de vídeo, que corresponde al componente de visualización de textura posterior a la codificación del componente de visualización de textura.

45 **[0166]** En algunos ejemplos, la unidad de procesamiento de predicción 42 del codificador de vídeo 20 puede ser un ejemplo de un procesador configurado para implementar los ejemplos descritos en esta divulgación para la obtención de vector de disparidad y la codificación de BVSP. En algunos ejemplos, una unidad (por ejemplo, uno o más procesadores) distinta a la unidad de procesamiento de predicción 42 puede implementar los ejemplos descritos anteriormente. En algunos ejemplos, la unidad de procesamiento de predicción 42, en conjunto con otras una o más unidades del codificador de vídeo 20, puede implementar los ejemplos descritos anteriormente. En algunos ejemplos, un procesador del codificador de vídeo 20 (no mostrado en la FIG. 8) puede, solo o en conjunto con otros procesadores del codificador de vídeo 20, implementar los ejemplos descritos anteriormente.

55 **[0167]** La FIG. 9 es un diagrama de bloques que ilustra un ejemplo de un decodificador de vídeo que puede implementar las técnicas descritas en esta divulgación. La FIG. 9 es un diagrama de bloques que ilustra un ejemplo de un decodificador de vídeo que puede implementar las técnicas descritas en esta divulgación. Por ejemplo, la FIG. 9, ilustra el decodificador de vídeo 30 que puede representar un decodificador de vídeo conforme a la norma 3D-AVC o a la norma 3D-HEVC. El decodificador de vídeo 30 se describirá usando cierta terminología de la HEVC, tal como PU, TU y CU, pero debería entenderse que las técnicas descritas con referencia al decodificador de vídeo 30 también pueden realizarse con vídeo codificado de acuerdo con la norma H.264.

60 **[0168]** El decodificador de vídeo 30 puede realizar la descodificación de predicción inter o la descodificación de predicción intra. La FIG. 9 ilustra el decodificador de vídeo 30. En el ejemplo de la FIG. 9, el decodificador de vídeo 30 incluye una memoria de datos de vídeo 69, una unidad de descodificación por entropía 70, una unidad de procesamiento de predicción 71, una unidad de procesamiento de cuantificación inversa 76, una unidad de procesamiento de transformada inversa 78, un sumador 80 y una memoria de imágenes de referencia 82. La unidad de procesamiento de predicción 71 incluye la unidad de compensación de movimiento y disparidad 72 y la unidad de

predicción intra 74. En algunos ejemplos, el descodificador de vídeo 30 puede realizar una pasada de descodificación que en general es recíproca a la pasada de codificación descrita con respecto al codificador de vídeo 20 de la FIG. 8.

5 **[0169]** La memoria de datos de vídeo 69 puede almacenar datos de vídeo, tales como un flujo de bits de vídeo codificado, a descodificar por parte de los componentes del descodificador de vídeo 30. Los datos de vídeo almacenados en la memoria de datos de vídeo 69 se pueden obtener, por ejemplo, a partir de un dispositivo de almacenamiento 34, a partir de un origen de vídeo local, tal como una cámara, mediante comunicación de red cableada o inalámbrica de datos de vídeo o accediendo a medios físicos de almacenamiento de datos. La memoria de datos de vídeo 69 puede formar un almacén temporal de imágenes codificadas (CPB) que almacene datos de vídeo codificados a partir de un flujo de bits de vídeo codificado.

15 **[0170]** La memoria de imágenes de referencia 82 es un ejemplo de una memoria intermedia de imágenes descodificadas (DPB) que almacena datos de vídeo de referencia para su uso en la descodificación de datos de vídeo por el descodificador de vídeo 30 (por ejemplo, en los modos de intra-codificación o inter-codificación). La memoria de datos de vídeo 69 y la memoria de imágenes de referencia 82 pueden ser formados por cualquiera entre varios dispositivos de memoria, tales como memoria dinámica de acceso aleatorio (DRAM), incluyendo DRAM síncrona (SDRAM), RAM magnetorresistiva (MRAM), RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. La memoria de datos de vídeo 69 y la memoria de imágenes de referencia 82 pueden ser proporcionadas por el mismo dispositivo de memoria o por dispositivos de memoria independientes. En diversos ejemplos, la memoria de datos de vídeo 69 puede estar en un chip con otros componentes del descodificador de vídeo 30, o fuera de chip con respecto a esos componentes.

25 **[0171]** Durante el proceso de descodificación, el descodificador de vídeo 30 recibe un flujo de bits de vídeo codificado, que representa bloques de vídeo de un fragmento de vídeo codificado y elementos 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 y otros elementos sintácticos. La unidad de descodificación por entropía 70 envía los vectores de movimiento y otros elementos sintácticos a la unidad de procesamiento de predicción 71. El descodificador de vídeo 30 puede recibir los elementos sintácticos en el nivel de fragmento de vídeo y/o el nivel de bloque de vídeo.

30 **[0172]** Cuando el fragmento de vídeo se codifica como un fragmento intra-codificado (I), la unidad de predicción intra 74 de la unidad de procesamiento de predicción 71 puede generar datos de predicción para un bloque de vídeo del fragmento de vídeo actual, basándose en un modo de predicción intra señalizada y en datos de bloques previamente descodificados de la trama o imagen actual. Cuando la trama de vídeo se codifica como un fragmento inter-codificado (es decir, B o P), la unidad de compensación de movimiento y disparidad 72 de la unidad de procesamiento de predicción 71 produce bloques predictivos para un bloque de vídeo del fragmento de vídeo actual, basándose en los vectores de movimiento y otros elementos sintácticos recibidos desde la unidad de descodificación por entropía 70. Los bloques predictivos se pueden generar a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El descodificador de vídeo 30 puede construir las listas de imágenes de referencia (RefPicList0 y RefPicList1) usando técnicas de construcción predeterminadas basándose en imágenes de referencia almacenadas en la memoria de imágenes de referencia 82.

45 **[0173]** La unidad de compensación de movimiento y disparidad 72 determina la información de predicción para un bloque de vídeo del fragmento de vídeo actual, analizando sintácticamente los vectores de movimiento y otros elementos sintácticos, y usa la información de predicción para generar los bloques predictivos para el bloque de vídeo actual que está siendo descodificado. Por ejemplo, la unidad de compensación de movimiento y disparidad 72 usa algunos de los elementos sintácticos recibidos para determinar un modo de predicción (por ejemplo, predicción intra o predicción inter), usada para codificar los bloques de vídeo del fragmento de vídeo, un tipo de fragmento de predicción inter (por ejemplo, fragmento B o fragmento P), información de construcción para una o más de las listas de imágenes de referencia para el fragmento, vectores de movimiento para cada bloque de vídeo inter-codificado del fragmento, el estado de predicción inter para cada bloque de vídeo inter-codificado del fragmento y otra información, para descodificar los bloques de vídeo en el fragmento de vídeo actual.

55 **[0174]** La unidad de compensación de movimiento y disparidad 72 también puede realizar la interpolación basándose en filtros de interpolación. La unidad de compensación de movimiento y disparidad 72 puede usar filtros de interpolación como los usados por el codificador de vídeo 20 durante la codificación de los bloques de vídeo, para calcular valores interpolados para píxeles fraccionarios de bloques de referencia. En este caso, la unidad de compensación de movimiento y disparidad 72 puede determinar los filtros de interpolación utilizados por el codificador de vídeo 20 a partir de los elementos sintácticos recibidos, y utilizar los filtros de interpolación para generar bloques predictivos.

60 **[0175]** La unidad de compensación de movimiento y disparidad 72 puede realizar una predicción de síntesis de visualización hacia atrás para un bloque actual en una visualización actual. Es decir, la unidad de compensación de movimiento y disparidad 72 puede determinar una imagen de la memoria de imágenes de referencia 82 en una primera visualización. Como se explicó con mayor detalle anteriormente, la unidad de compensación de movimiento y disparidad 72 puede determinar un bloque de profundidad correspondiente al bloque actual, y al usar los valores de

profundidad del bloque de profundidad, la unidad de compensación de movimiento y disparidad 72 puede distorsionar los valores de píxeles de una imagen en la primera visualización, determinada en relación con las posiciones de los píxeles en el bloque actual, de manera que un bloque predicho (es decir, un bloque de referencia BVSP) se forma en una segunda visualización, diferente de la primera visualización y la visualización actual. La unidad de compensación de movimiento y disparidad 72 puede proporcionar este bloque previsto para el sumador 50 y el sumador 80, para su uso en el cálculo de un residuo y en la reproducción del bloque actual, respectivamente. Del mismo modo, de acuerdo con las técnicas de esta divulgación, el descodificador de vídeo 30 puede descodificar datos de sintaxis que definen información de movimiento para el bloque actual de tal manera que la información de movimiento incluye un índice de referencia que tiene un valor que identifica la imagen en la primera visualización desde la cual el bloque predicho se sintetiza.

[0176] La unidad de procesamiento de cuantificación inversa 76 realiza la cuantificación 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 proceso de cuantificación inversa puede incluir el uso de un parámetro de cuantificación calculado por el codificador de vídeo 20 para cada bloque de vídeo del fragmento de vídeo con el fin de determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa que se debería aplicar. La unidad de procesamiento de transformada inversa 78 aplica una transformada inversa (por ejemplo, una DCT inversa, una transformada inversa entera o un proceso de transformada inversa conceptualmente similar) a los coeficientes de transformada, a fin de producir bloques residuales en el dominio de píxeles.

[0177] Después de que la unidad de compensación de movimiento y disparidad 72 genera el bloque predictivo para el bloque de vídeo actual, basándose en los vectores de movimiento y otros elementos sintácticos, el descodificador de vídeo 30 forma un bloque de vídeo descodificado sumando los bloques residuales procedentes de la unidad de procesamiento de transformada inversa 78 a los correspondientes bloques predictivos generados por la unidad de compensación de movimiento y disparidad 72. El sumador 80 representa el componente o los componentes que realizan esta operación de suma. Si se desea, también se puede aplicar un filtro de eliminación de bloques para filtrar los bloques descodificados a fin de eliminar distorsiones de efecto pixelado. También se pueden usar otros filtros de bucle (en el bucle de codificación o bien después del bucle de codificación) para allanar las transiciones de píxeles o mejorar de otro modo la calidad del vídeo. Los bloques de vídeo descodificados en una imagen dada se almacenan a continuación en la memoria de imágenes de referencia 82, que almacena imágenes de referencia usadas para la posterior compensación de movimiento. La memoria de imágenes de referencia 82 almacena también vídeo descodificado para su presentación posterior en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la FIG. 1.

[0178] De esta manera, el descodificador de vídeo 30 es un ejemplo de un descodificador de vídeo que puede configurarse para implementar una o más técnicas a modo de ejemplo descritas en esta divulgación. Por ejemplo, la memoria de datos de vídeo 69 almacena datos de vídeo. Los datos de vídeo pueden incluir información a partir de la cual el descodificador de vídeo 30 puede descodificar un componente de vídeo de textura de una visualización dependiente y un componente de visualización de profundidad que corresponde al componente de visualización de textura, cada uno de los cuales codificó el codificador de vídeo 20 en un proceso de codificación de vídeo conforme a la norma 3D-AVC o conforme a la norma 3D-HEVC.

[0179] En las técnicas descritas en esta divulgación, el descodificador de vídeo 30 puede incluir uno o más procesadores que están configurados para descodificar, en un proceso de codificación de vídeo conforme a la norma 3D-AVC o conforme a la norma 3D-HEVC, un componente de visualización de textura de una visualización dependiente de los datos de vídeo. Para descodificar el componente de visualización de textura, el descodificador de vídeo 30 se puede configurar para evaluar la información de movimiento de uno o más bloques contiguos de un bloque actual en el componente de visualización de textura, para determinar si al menos un bloque contiguo está predicho entre visualizaciones con un vector de movimiento de disparidad que se refiere a una imagen de referencia entre visualizaciones en una visualización distinta a la visualización dependiente. El descodificador de vídeo 30 puede obtener un vector de disparidad para el bloque actual basándose en el vector de movimiento de disparidad para uno de los bloques contiguos. Para la codificación de textura en primer lugar, el descodificador de vídeo 30 puede descodificar un componente de visualización de profundidad, de los datos de vídeo, que corresponde al componente de visualización de textura posterior a la descodificación del componente de visualización de textura.

[0180] En algunos ejemplos, la unidad de procesamiento de predicción 71 del descodificador de vídeo 30 puede ser un ejemplo de un procesador configurado para implementar los ejemplos descritos en esta divulgación para la obtención de vectores de disparidad y la codificación de BVSP. En algunos ejemplos, una unidad (por ejemplo, uno o más procesadores) distinta a la unidad de procesamiento de predicción 71, puede implementar los ejemplos descritos anteriormente. En algunos ejemplos, la unidad de procesamiento de predicción 71, en conjunto con otras una o más unidades del descodificador de vídeo 30, pueden implementar los ejemplos descritos anteriormente. En algunos otros ejemplos más, un procesador del descodificador de vídeo 30 (no mostrado en la FIG. 9) puede, solo o en conjunto con otros procesadores del descodificador de vídeo 30, implementar los ejemplos descritos anteriormente.

[0181] La FIG. 10 es un diagrama de flujo que muestra un ejemplo de proceso de codificación de vídeo 30, de acuerdo con las técnicas de esta divulgación. Las técnicas de la FIG. 10 puede ser aplicables a vídeo conforme con

3D-AVC o a vídeo conforme con 3D-HEVC. Las técnicas de la FIG. 10 se describirán con respecto a un codificador de vídeo genérico, que puede ser, por ejemplo, un codificador de vídeo como el codificador de vídeo 20 o un descodificador de vídeo como el descodificador de vídeo. El codificador de vídeo determina que un bloque de una primera visualización de textura debe codificarse utilizando un modo BVSP (110). Al realizar la descodificación de vídeo, el descodificador de vídeo 30 puede, por ejemplo, determinar que el bloque de la primera visualización de textura debe descodificarse usando el modo BVSP al recibir un elemento sintáctico que indica que el bloque de la primera visualización de textura se debe codificar usando el modo de síntesis de visualización basado en bloques. Al realizar la codificación de vídeo, el codificador de vídeo 20 puede, por ejemplo, determinar que el bloque de la primera visualización de textura se va a codificar utilizando el modo BVSP realizando varias pasadas de codificación e identificando el modo BVSP como un modo que produce una compensación de velocidad-distorsión deseada.

[0182] El codificador de vídeo ubica, en una visualización de profundidad, un bloque de profundidad que corresponde al bloque de la primera visualización de textura (112). En 3D-AVC, el bloque en este ejemplo puede referirse a una sub-región KxK de un macrobloque o partición de macrobloque. En 3D-HEVC, el bloque también puede referirse a una sub-región KxK. El bloque de profundidad correspondiente puede, por ejemplo, ser un bloque de profundidad co-ubicado en 3D-AVC o puede ser un bloque de profundidad en la visualización básica de referencia (es decir, la segunda visualización) que se identifica mediante el vector de disparidad generado usando NBDV en 3D-HEVC. El codificador de vídeo determina los valores de profundidad para dos o más posiciones de esquina del bloque de profundidad (114). Basándose en los valores de profundidad, el codificador de vídeo obtiene un vector de disparidad para el bloque (116). Usando el vector de disparidad, el codificador de vídeo ubica un bloque de una segunda visualización de textura (118). El codificador de vídeo realiza una predicción inter del bloque de la primera visualización de textura usando el bloque de la segunda visualización de textura (120). La primera visualización de textura puede ser, por ejemplo, una visualización de textura no básica, y la segunda visualización de textura puede ser, por ejemplo, una visualización de textura básica. Cuando las técnicas de la FIG. 10 se implementan mediante un codificador de vídeo conforme con 3D-AVC, el bloque de la primera visualización de textura puede ser, por ejemplo, un sub-bloque de una partición de macrobloque. Cuando las técnicas de la FIG. 10 se implementan mediante un codificador compatible con 3D-HEVC, el bloque de la primera visualización de textura puede, por ejemplo, ser una unidad de predicción.

[0183] La FIG. 11 es un diagrama de flujo que muestra un ejemplo de proceso de codificación de vídeo 30, de acuerdo con las técnicas de esta divulgación. Las técnicas de la FIG. 11 se describirá utilizando la terminología 3D-AVC, pero las técnicas podrían extenderse a otras normas de codificación de vídeo como 3D-HEVC. Las técnicas de la FIG. 11 se describirán con respecto a un codificador de vídeo genérico, que puede ser, por ejemplo, un codificador de vídeo como el codificador de vídeo 20 o un descodificador de vídeo como el descodificador de vídeo. El codificador de vídeo realiza la primera codificación de textura para una primera visualización de textura, una primera visualización de profundidad, una segunda visualización de textura y una segunda visualización de profundidad (122). La primera visualización de textura y la primera visualización de profundidad básica pueden ser, por ejemplo, visualizaciones básicas, mientras que la segunda visualización de textura y la segunda visualización de profundidad son visualizaciones no básicas.

[0184] Para un macrobloque de la segunda visualización de textura, el codificador de vídeo ubica un bloque de profundidad, en la primera visualización de profundidad, que corresponde al macrobloque (124). Basado en al menos un valor de profundidad del bloque de profundidad, el codificador de vídeo obtiene un vector de disparidad para el macrobloque (126). El codificador de vídeo puede, por ejemplo, obtener el vector de disparidad al determinar un conjunto de valores de profundidad que incluye valores de profundidad de dos o más muestras de esquina del bloque de profundidad, e identificar, a partir del conjunto de valores de profundidad, un valor de profundidad máxima. El valor de profundidad máximo se puede convertir luego en un vector de disparidad basado en una tabla de conversión que convierte los valores de profundidad en valores de disparidad o utilizando alguna otra técnica. El codificador de vídeo codifica un primer sub-bloque del macrobloque basado en el vector de disparidad obtenido (128). El codificador de vídeo codifica un segundo sub-bloque del macrobloque basado en el vector de disparidad obtenido (130). El codificador de vídeo puede, por ejemplo, codificar el primer sub-bloque usando uno de un modo de salto y un modo directo y codificar el segundo sub-bloque usando un modo de predicción inter diferente al modo de salto o el modo directo. En el ejemplo de la FIG. 11, los sub-bloques pueden ser, por ejemplo, sub-bloques H.264/AVC o pueden ser particiones de macrobloque.

[0185] En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, firmware o cualquier combinación de los mismos. Si se implementan en software, las funciones se pueden almacenar en, o transmitir por, un medio legible por ordenador, como una o más instrucciones o código, y ejecutarse mediante una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que correspondan a un medio tangible tal como medios de almacenamiento de datos, o medios de comunicación que incluyan cualquier medio que facilite la transferencia de un programa informático desde un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador pueden corresponder en general a (1) medios de almacenamiento tangibles legibles por ordenador que sean no transitorios o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser medios disponibles cualesquiera a los que se pueda acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para

la implementación de las técnicas descritas en esta divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

5 **[0186]** A modo de ejemplo, y no de limitación, tales medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento de disco óptico, almacenamiento de disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que pueda usarse para almacenar código de programa deseado en forma de instrucciones o estructuras de datos y al que pueda accederse mediante un ordenador. Además, cualquier conexión recibe adecuadamente la denominación de medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde una sede de la Red, un servidor u otro origen remoto usando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas se incluyen en la definición de medio. Sin embargo, debería entenderse que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que, en cambio, se orientan a medios de almacenamiento tangibles no transitorios. El término disco, como se usa en el presente documento, incluye 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 habitualmente emiten datos magnéticamente, mientras que otros discos emiten datos ópticamente con láseres. Las combinaciones de lo anterior también deberían incluirse dentro del alcance de los medios legibles por ordenador.

20 **[0187]** Las instrucciones pueden ser ejecutadas por uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), formaciones de puertas programables in situ (FPGA) u otros circuitos lógicos, integrados o discretos, equivalentes. En consecuencia, el término "procesador", como se usa en el presente documento, se puede referir a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento se puede proporcionar dentro de módulos de hardware y/o software dedicados, configurados para la codificación y la descodificación, o incorporados en un códec combinado. Además, las técnicas se podrían implementar totalmente en uno o más circuitos o elementos lógicos.

30 **[0188]** Las técnicas de la presente divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluidos un equipo manual inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). Diversos componentes, módulos o unidades se describen en esta divulgación para enfatizar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no requieren necesariamente su realización mediante diferentes unidades de hardware. En cambio, como se ha descrito anteriormente, diversas unidades se pueden combinar en una unidad de hardware de códec, o ser proporcionadas por un grupo de unidades de hardware interoperativas, incluyendo uno o más procesadores, como se ha descrito anteriormente, conjuntamente con software y/o firmware adecuados.

40 **[0189]** 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 tridimensionales (3D), comprendiendo el procedimiento:
- 5 la determinación de un primer macrobloque de una primera visualización de textura debe descodificarse utilizando un modo de síntesis de visualización basada en bloques;
- 10 ubicar, en una visualización de profundidad, un primer bloque de profundidad que corresponde al primer macrobloque de la primera visualización de textura;
- determinar los valores de profundidad de dos o más posiciones de esquina del primer bloque de profundidad;
- 15 basándose en los valores de profundidad, obtener un vector de disparidad para el primer macrobloque, **caracterizado por:**
- 20 para un primer sub-bloque del primer macrobloque de la primera visualización de textura, descodificar usando el vector de disparidad;
- 25 para un segundo sub-bloque del primer macrobloque de la primera visualización de textura, descodificar usando el vector de disparidad;
- determinar un segundo macrobloque de la primera visualización de textura debe descodificarse utilizando un modo seleccionado del grupo que consiste en un modo de salto, un modo directo, un modo de fusión y un modo de predicción inter;
- 30 ubicar, en la visualización de profundidad, un segundo bloque de profundidad que corresponde al segundo macrobloque de la primera visualización de textura;
- 35 determinar los valores de profundidad de dos o más posiciones de esquina del segundo bloque de profundidad, en el que las dos o más posiciones de esquina del segundo bloque de profundidad comprenden las mismas dos o más posiciones de esquina del segundo bloque de profundidad como el primer bloque de profundidad;
- 40 basándose en los valores de profundidad del segundo bloque de profundidad, obtener un vector de disparidad para el segundo macrobloque;
- para un primer sub-bloque del segundo macrobloque de la primera visualización de textura, descodificar usando el vector de disparidad para el segundo macrobloque;
- 45 para un segundo sub-bloque del segundo macrobloque de la primera visualización de textura, descodificar usando el vector de disparidad para el segundo macrobloque.
2. El procedimiento según la reivindicación 1, en el que la primera visualización de textura comprende una visualización de textura no básica y en el que la segunda visualización de textura comprende una visualización de textura básica.
3. El procedimiento según la reivindicación 1, en el que el primer bloque de la primera visualización de textura comprende un bloque KxK y en el que el primer bloque de profundidad que corresponde al bloque de la primera visualización de textura comprende uno de un bloque K/2xK/2 o un bloque KxK.
4. El procedimiento según la reivindicación 1, en el que el primer macrobloque de la primera visualización de textura se codifica de acuerdo con la norma de codificación de vídeo 3D-AVC, y la segunda visualización de textura se codifica de acuerdo con la norma de codificación de vídeo H.264/AVC y el primer bloque de la primera visualización de textura está contenido dentro de una partición de macrobloque.
5. El procedimiento según la reivindicación 1, en el que el primer bloque se codifica de acuerdo con la norma de codificación de vídeo de alta eficiencia (HEVC) 3D; y la segunda visualización de textura se codifica de acuerdo con la norma HEVC y el primer bloque de la primera visualización de textura está contenido dentro de una unidad de predicción.
6. El procedimiento según la reivindicación 1, en el que obtener un vector de disparidad para el primer macrobloque comprende determinar un valor de profundidad máximo a partir de los valores de profundidad de dos o más posiciones de esquina del bloque de profundidad.

7. El procedimiento según la reivindicación 1, en el que el procedimiento de codificación de datos de vídeo comprende un procedimiento de descodificación de datos de vídeo, y en el que determinar si el primer bloque de la primera visualización de textura se va a codificar utilizando el modo de síntesis de visualización basado en bloques comprende recibir un elemento sintáctico, en el que el elemento sintáctico indica que el primer bloque de la primera visualización de textura debe codificarse utilizando el modo de síntesis de la visualización basada en bloques.
8. Un dispositivo para descodificar datos de vídeo tridimensionales (3D), comprendiendo el dispositivo:
- medios para determinar un primer macrobloque de una primera visualización de textura debe codificarse utilizando un modo de síntesis de visualización basada en bloques;
 - medios para ubicar, en una visualización en profundidad, un primer bloque de profundidad que corresponde al primer macrobloque de la primera visualización de textura;
 - medios para determinar valores de profundidad de dos o más posiciones de esquina del primer bloque de profundidad;
 - basado en los valores de profundidad, medios para obtener un vector de disparidad para el primer macrobloque, **caracterizado por:**
 - para un primer sub-bloque del primer macrobloque de la primera visualización de textura, medios para descodificar usando el vector de disparidad;
 - para un segundo sub-bloque del primer macrobloque de la primera visualización de textura, medios para descodificar usando el vector de disparidad;
 - los medios para determinar un segundo macrobloque de la primera visualización de textura se descodificarán utilizando un modo seleccionado del grupo que consiste en un modo de salto, un modo directo, un modo de fusión y un modo de predicción inter;
 - medios para ubicar, en la visualización de profundidad, un segundo bloque de profundidad que corresponde al segundo macrobloque de la primera visualización de textura;
 - medios para determinar los valores de profundidad de dos o más posiciones de esquina del segundo bloque de profundidad, en el que las dos o más posiciones de esquina del segundo bloque de profundidad comprenden las mismas dos o más posiciones de esquina del segundo bloque de profundidad como el primer bloque de profundidad;
 - basado en los valores de profundidad del segundo bloque de profundidad, medios para obtener un vector de disparidad para el segundo macrobloque;
 - para un primer sub-bloque del segundo macrobloque de la primera visualización de textura, medios para descodificar usando el vector de disparidad para el segundo macrobloque;
 - para un segundo sub-bloque del segundo macrobloque de la primera visualización de textura, medios para descodificar usando el vector de disparidad para el segundo macrobloque.
9. El dispositivo de la reivindicación 8, en el que la primera visualización de textura comprende una visualización de textura no básica y en el que la segunda visualización de textura comprende una visualización de textura básica.
10. El dispositivo de la reivindicación 9, en el que el primer macrobloque de la primera visualización de textura se codifica de acuerdo con la norma de codificación de vídeo 3D-AVC, y la segunda visualización de textura se codifica de acuerdo con la norma de codificación de vídeo H.264/AVC, y en el que el primer macrobloque de la primera visualización de textura está contenido dentro de una partición de macrobloque.
11. El dispositivo de la reivindicación 8, en el que el primer macrobloque está codificado de acuerdo con la norma de codificación de vídeo de alta eficiencia en 3D (HEVC); y la segunda visualización de textura se codifica de acuerdo con la norma HEVC, y en el que el primer macrobloque de la primera visualización de textura está contenido dentro de una unidad de predicción.
12. Un medio de almacenamiento legible por ordenador que almacena instrucciones que, cuando son ejecutadas por uno o más procesadores, hacen que el uno o más procesadores realicen el procedimiento de cualquiera de las reivindicaciones 1 a 7.

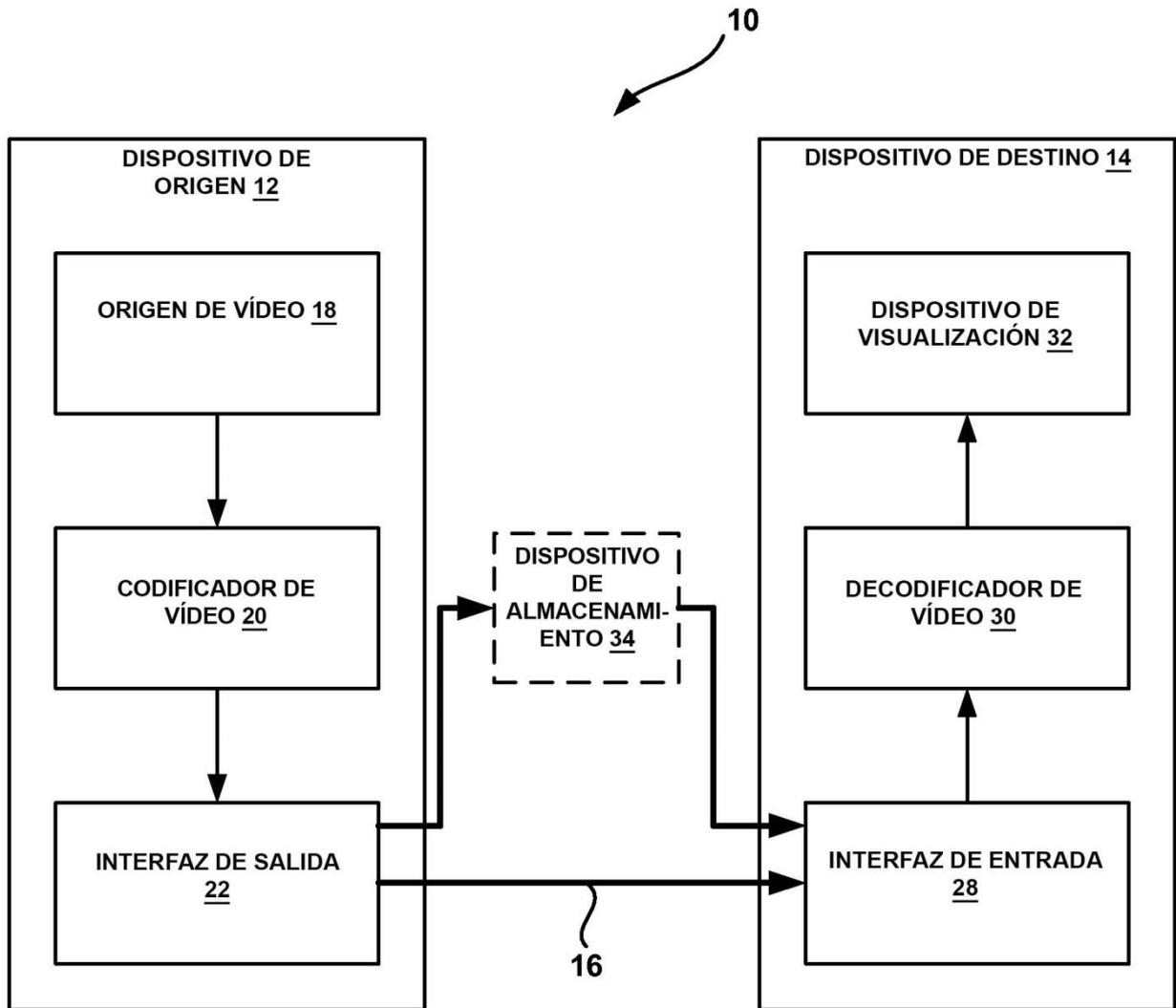


FIG. 1

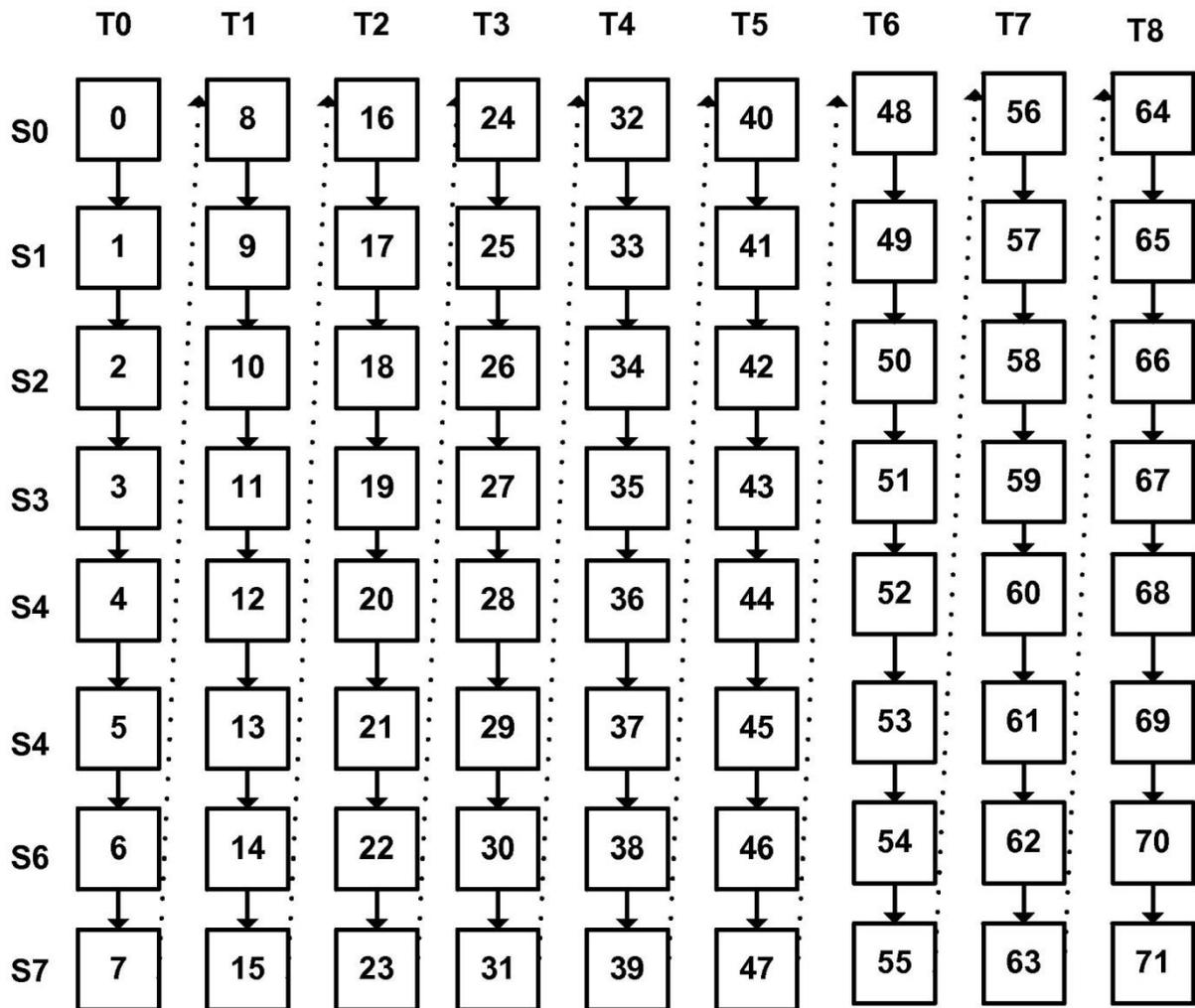


FIG. 2

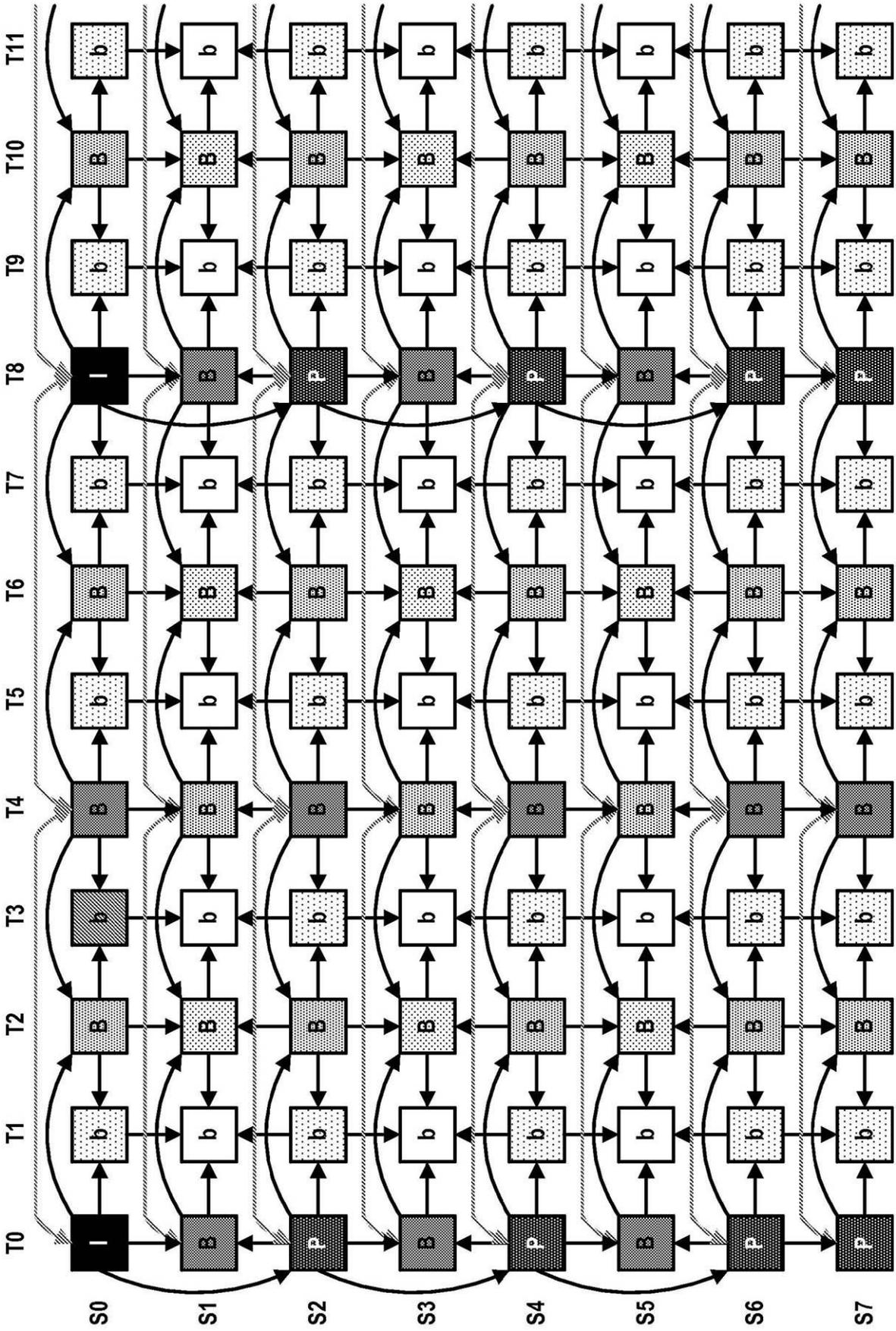


FIG. 3

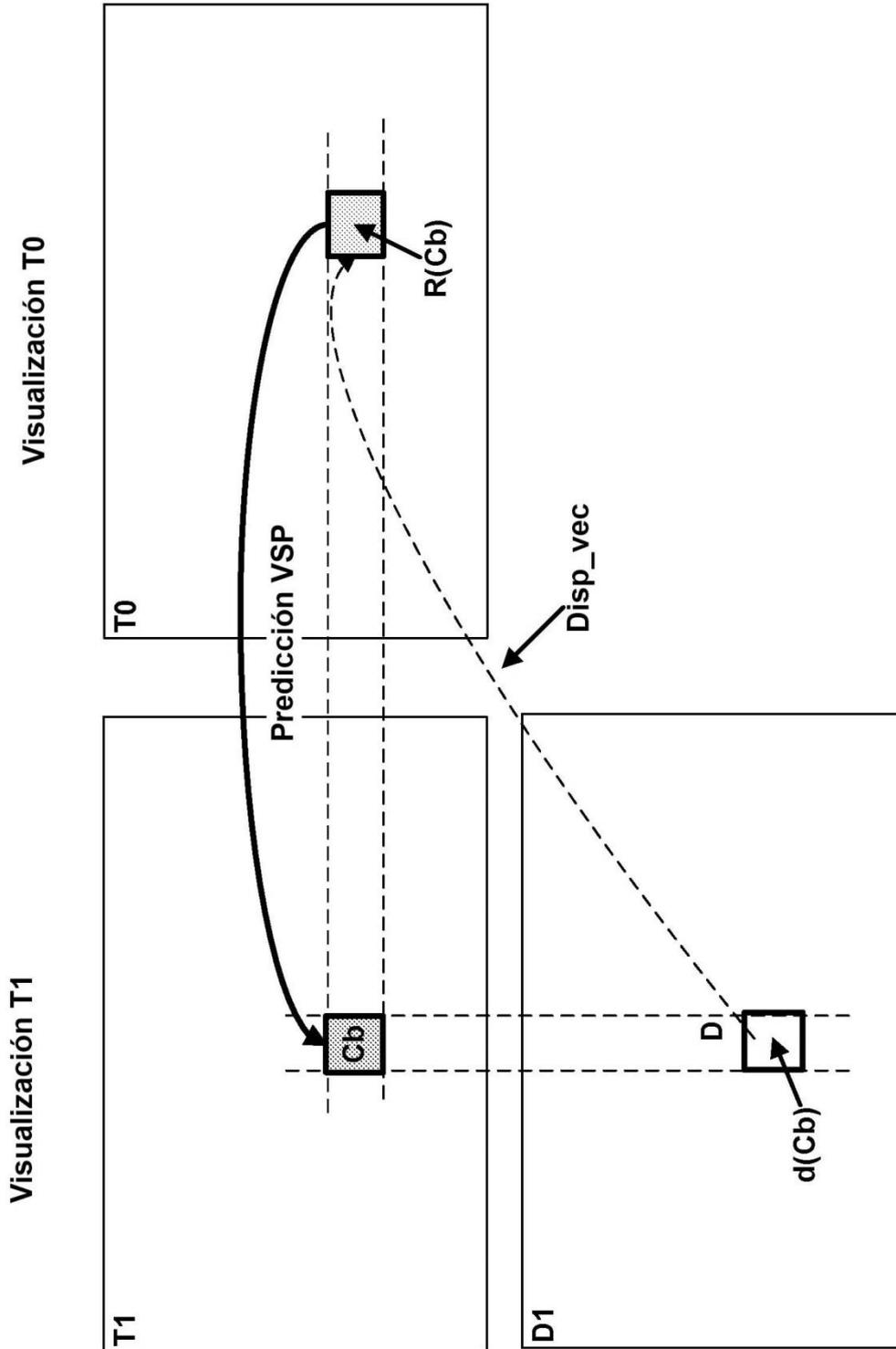


FIG. 4

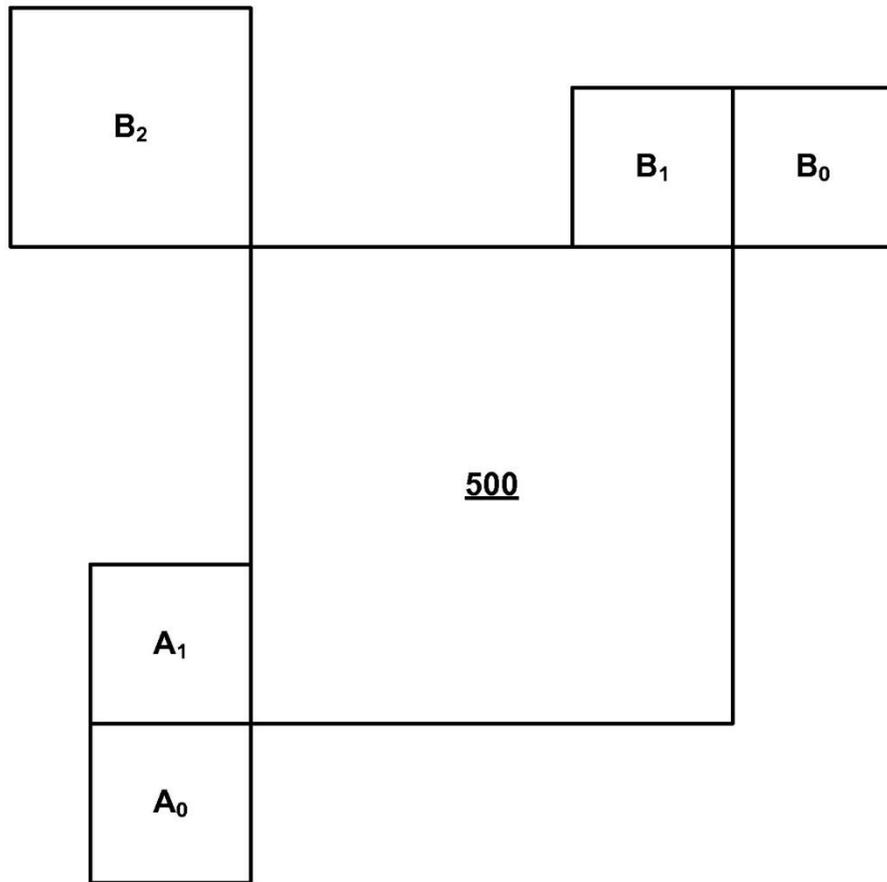


FIG. 5

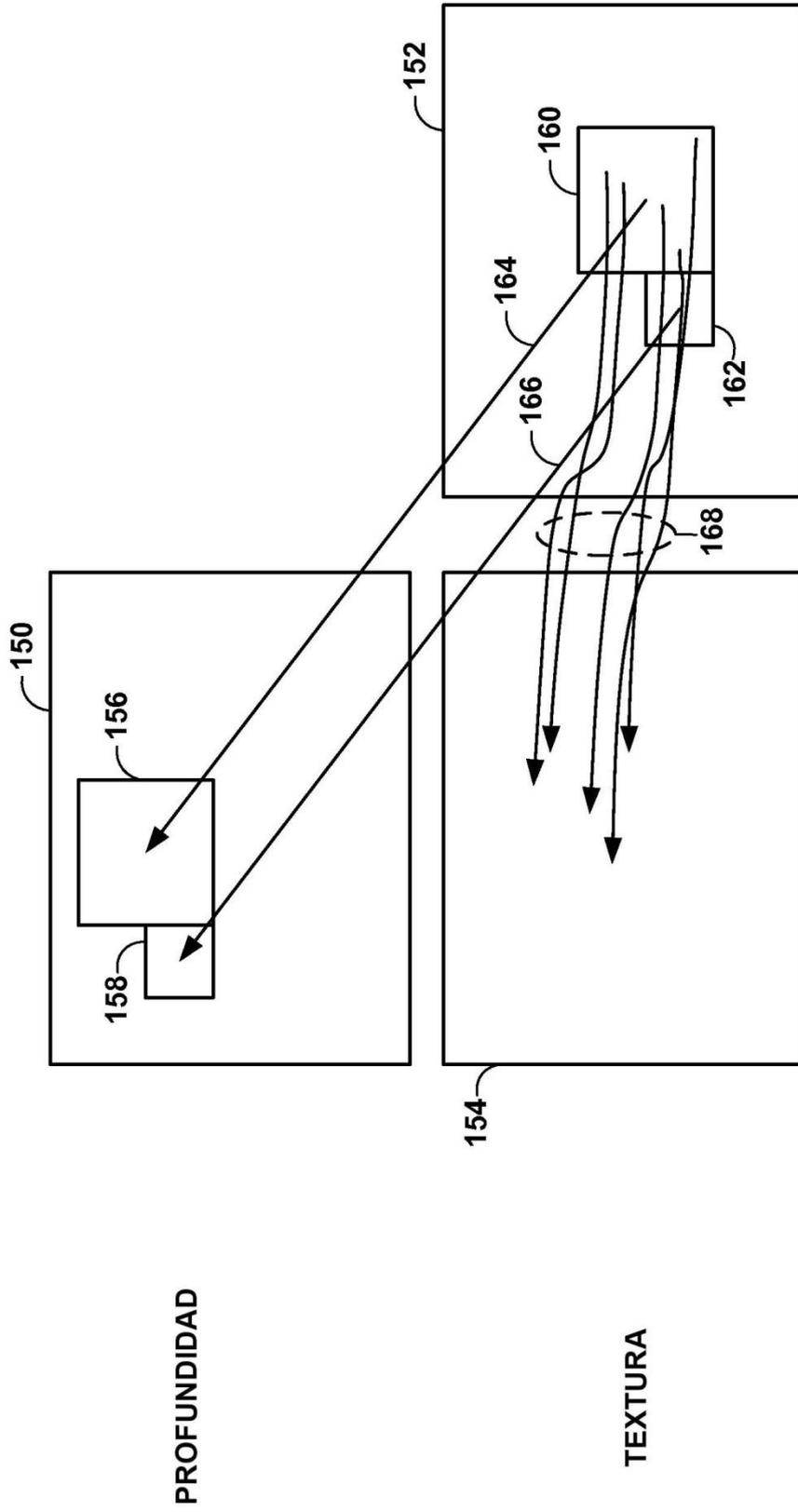


FIG. 6

170

| | | | | | | | |
|-------------|--|--|--|--|--|--|-------------|
| 172C | | | | | | | 172D |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| 172A | | | | | | | 172B |

FIG. 7

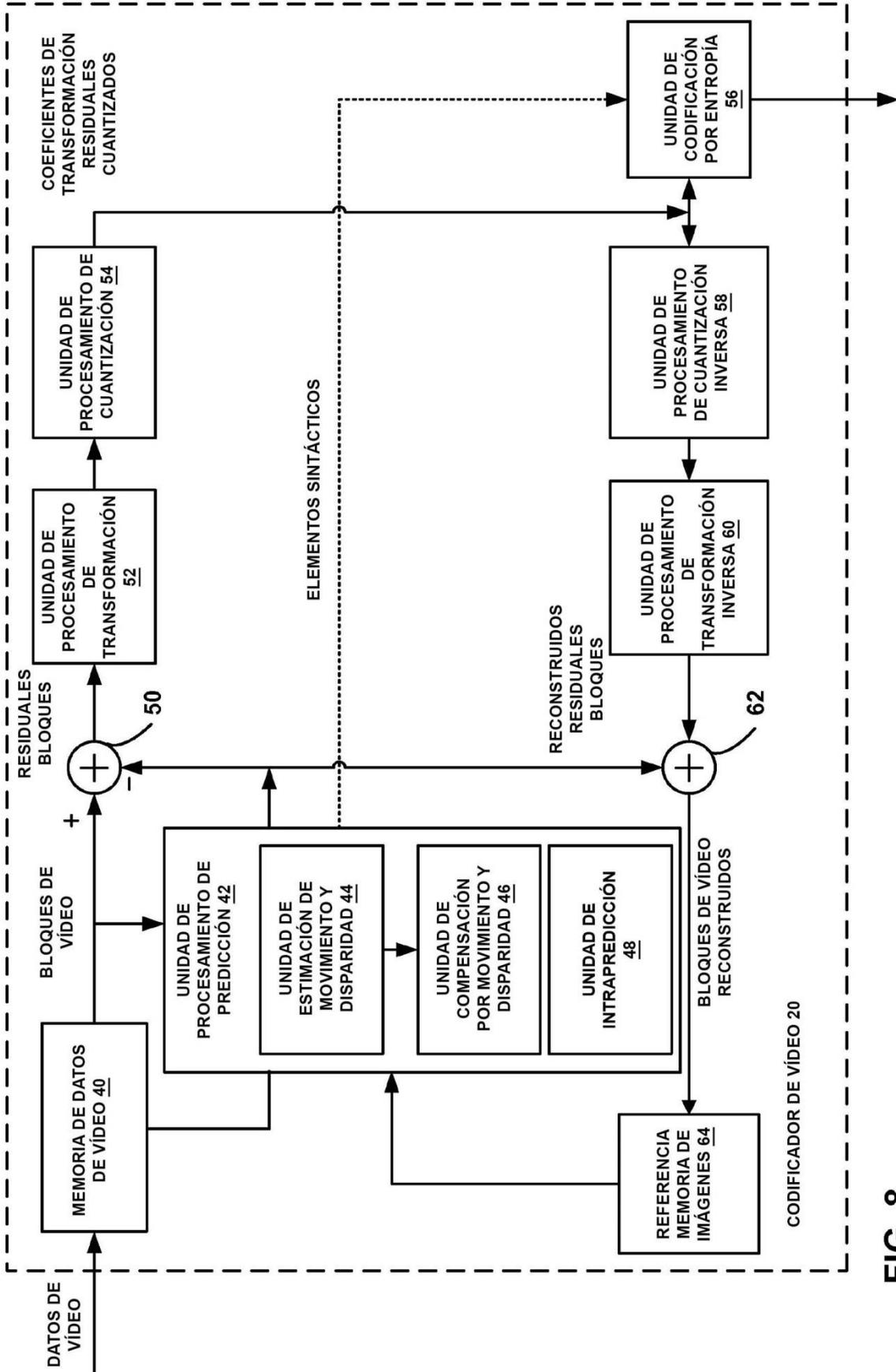


FIG. 8

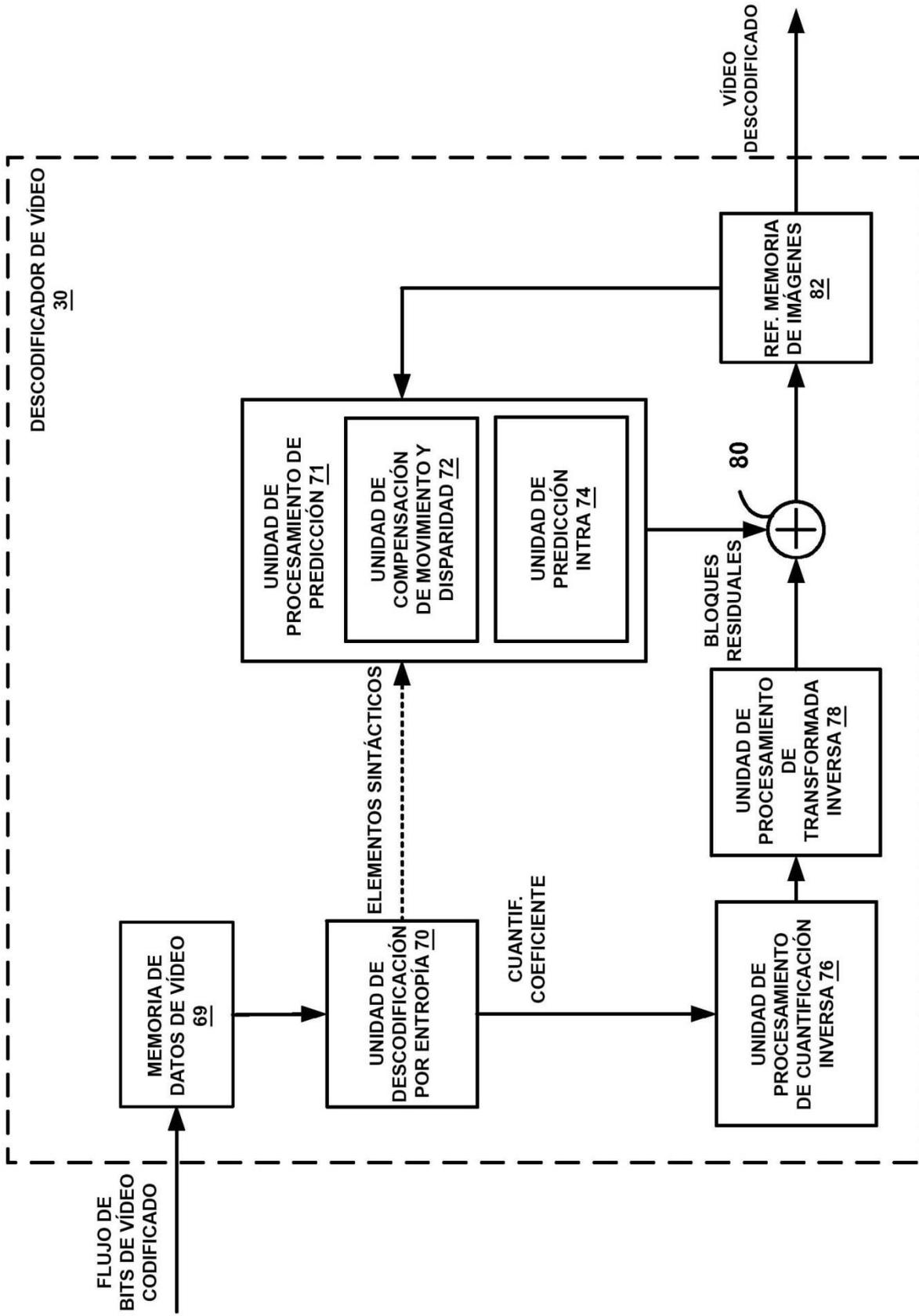


FIG. 9

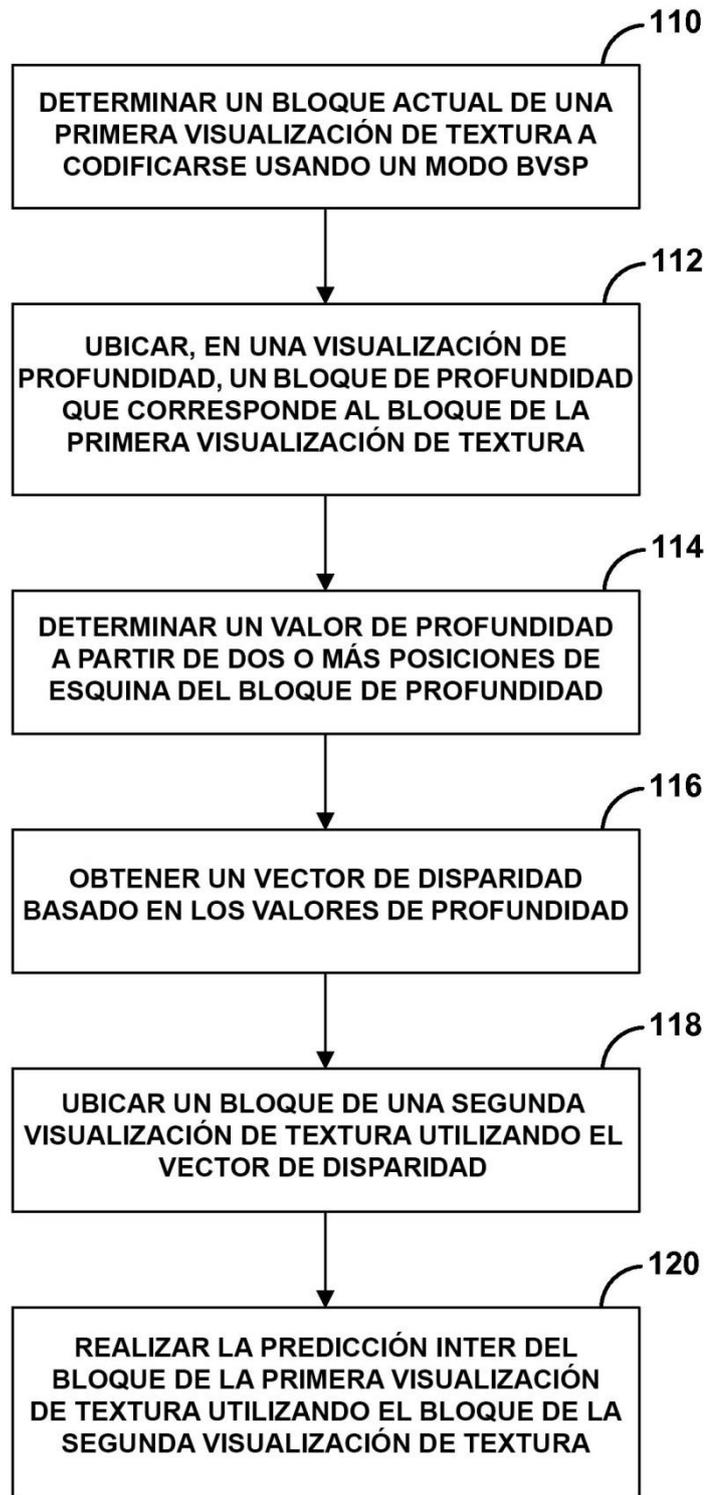


FIG. 10

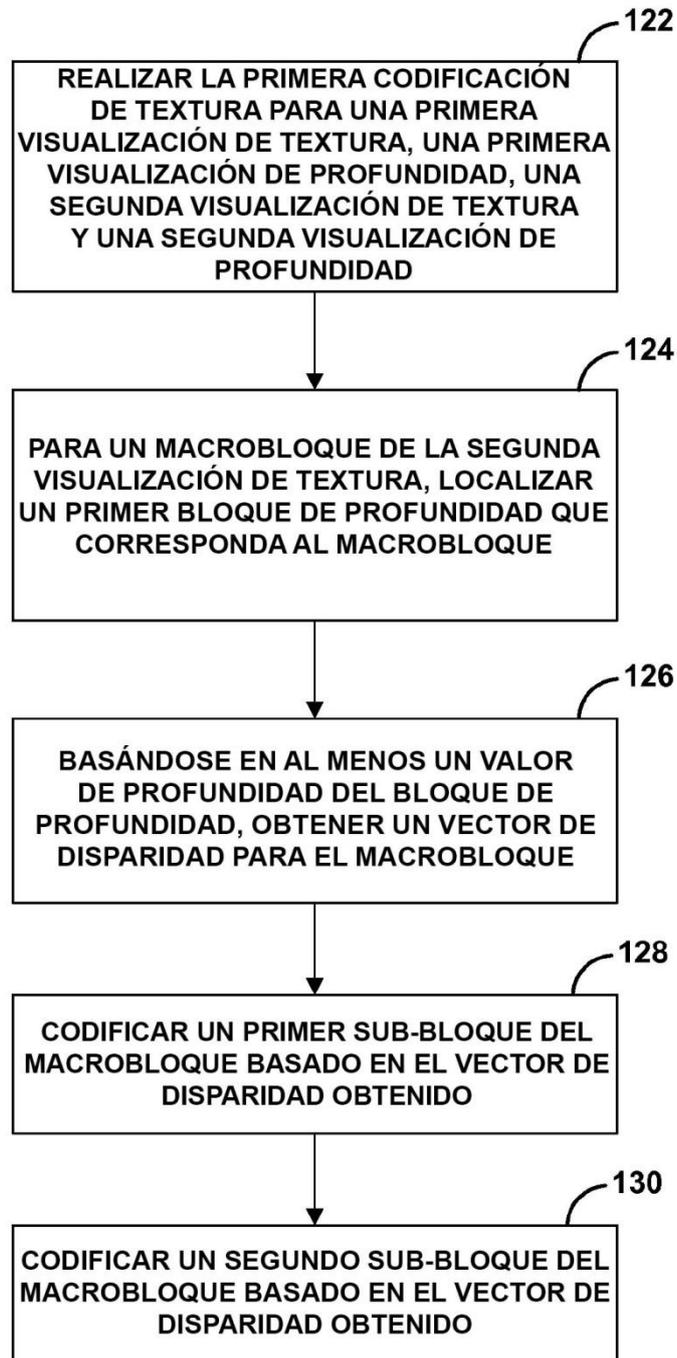


FIG. 11