

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 634 100**

51 Int. Cl.:

H04N 19/597 (2014.01)
H04N 19/172 (2014.01)
H04N 19/174 (2014.01)
H04N 19/44 (2014.01)
H04N 19/46 (2014.01)
H04N 19/61 (2014.01)
H04N 19/70 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **20.07.2012 PCT/US2012/047705**
 87 Fecha y número de publicación internacional: **31.01.2013 WO13016233**
 96 Fecha de presentación y número de la solicitud europea: **20.07.2012 E 12745946 (9)**
 97 Fecha y número de publicación de la concesión europea: **03.05.2017 EP 2735155**

54 Título: **Extensión de vídeo tridimensional de cabecera de fragmento para predicción de cabecera de fragmento**

30 Prioridad:

22.07.2011 US 201161510738 P
11.08.2011 US 201161522584 P
26.11.2011 US 201161563772 P
06.02.2012 US 201261595612 P
13.04.2012 US 201261624031 P
23.04.2012 US 201261637212 P
19.07.2012 US 201213553628

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
26.09.2017

73 Titular/es:

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

72 Inventor/es:

CHEN, YING;
WANG, YE-KUI y
KARCZEWICZ, MARTA

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 634 100 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

Extensión de vídeo tridimensional de cabecera de fragmento para predicción de cabecera de fragmento

- 5 La presente solicitud reivindica el beneficio de las solicitudes provisionales de Estados Unidos n.º 61/510 738, presentada el 22 de julio de 2011, n.º 61/522 584, presentada el 11 de agosto de 2011, n.º 61/563 772, presentada el 26 de noviembre de 2011, n.º 61/595 612, presentada el 6 de febrero de 2012, n.º 61/624 031, presentada el 13 de abril de 2012 y n.º 61/637 212, presentada el 23 de abril de 2012.

10 **CAMPO TÉCNICO**

La presente divulgación se refiere al campo de la codificación de vídeo, por ejemplo, la codificación de datos de vídeo tridimensional.

15 **ANTECEDENTES**

- Las capacidades de vídeo digital pueden incorporarse a una amplia gama de dispositivos, incluidos televisores digitales, sistemas de radiodifusión directa digital, dispositivos de comunicación inalámbrica, tales como radiotelefonos, sistemas de radiodifusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, cámaras digitales, dispositivos de grabación digital, dispositivos de videojuegos, consolas de videojuegos y similares. Los dispositivos de vídeo digital implementan técnicas de compresión de vídeo, tales como las de las normas MPEG-2, MPEG-4 o H.264/MPEG-4, parte 10, Codificación avanzada de vídeo (AVC), para transmitir y recibir información de vídeo digital de manera más eficiente. Las técnicas de compresión de vídeo llevan a cabo una predicción espacial y temporal para reducir o eliminar la redundancia inherente a las secuencias de vídeo.

- Las técnicas de compresión de vídeo llevan a cabo una predicción espacial y/o una predicción temporal para reducir o eliminar la redundancia inherente a las secuencias de vídeo. Para la codificación de vídeo basada en bloques, una trama o un fragmento de vídeo pueden dividirse en macrobloques. Cada macrobloque se puede dividir aún más. Los macrobloques de una trama o un fragmento intracodificado (I) se codifican mediante predicción espacial con respecto a los macrobloques contiguos. Los macrobloques de una trama o fragmento intercodificado (P o B) pueden utilizar predicción espacial con respecto a los macrobloques contiguos de la misma trama o fragmento, o predicción temporal con respecto a otras tramas de referencia.

- Después de que se hayan codificado los datos de vídeo, los datos de vídeo pueden agruparse en paquetes para su transmisión o almacenamiento. Los datos pueden reunirse en un archivo de vídeo correspondiente a cualquiera de una variedad de normas, tales como el formato de archivos de medios básico de la Organización internacional de normalización (ISO) y extensiones del mismo, tales como la AVC.

- Se han dedicado esfuerzos para elaborar nuevas normas de codificación de vídeo basándose en H.264/AVC. Una de estas normas es la norma de codificación de vídeo escalable (SVC), que es la extensión escalable para H.264/AVC. Otra norma es la codificación de vídeo multivista (MVC), que se ha convertido en la extensión multivista para H.264/AVC. Un proyecto conjunto de la MVC se describe en el documento JVT-AB204, «Joint Draft 8.0 on Multiview Video Coding», 28th JVT meeting, Hannover, Alemania, julio de 2008, disponible en http://wftp3.itu.int/av-arch/jvt-site/2008_07_Hannover/JVT-AB204.zip. Una versión de la norma AVC se describe en el documento JVT-AD007, «Editors' draft revision to UIT-T Rec. H.264 | ISO/IEC 14496-10 Advanced Video Coding - in preparation for UIT-T SG 16 AAP Consent (in integrated form)», 30th JVT meeting, Ginebra, Suiza, febrero de 2009, disponible en http://wftp3.itu.int/av-arch/jvt-site/2009_01_Geneva/JVT-AD007.zip. Este documento integra la SVC y la MVC en la especificación de la AVC. El documento de YING CHEN et al.: «Description of 3D video coding technology proposal by Qualcomm Incorporated», 98, MPEG MEETING; 28-11-2011 - 2-12-2011; GINEBRA; (GRUPO DE EXPERTOS DE IMÁGENES EN MOVIMIENTO o ISO/IEC JTC1/SC29/WG11), n.º m22583, 21 de noviembre de 2011 (21-11-2011), XP030051146, divulga un procedimiento de codificación de datos de vídeo. El procedimiento comprende codificar uno o más bloques de datos de vídeo representativos de información de textura de al menos una parte de una trama de los datos de vídeo; procesar un fragmento de textura para un componente de vista de textura de una vista actual asociada con una unidad de acceso, comprendiendo el fragmento de textura el uno o más bloques codificados y una cabecera de fragmento de textura que comprende un conjunto de elementos sintácticos representativos de unas características del fragmento de textura, codificar información de profundidad representativa de valores de profundidad para al menos la parte de la trama; y procesar un fragmento de profundidad para un componente de vista de profundidad de la vista actual correspondiente al componente de vista de textura de la vista actual, comprendiendo el fragmento de profundidad la información de profundidad codificada y una cabecera de fragmento de profundidad que comprende un conjunto de elementos sintácticos representativos de unas características del fragmento de profundidad, en el que procesar el fragmento de profundidad comprende predecir al menos una de las etapas de generar o recibir el fragmento de profundidad; en el que procesar el fragmento de profundidad comprende predecir al menos un elemento sintáctico del conjunto de elementos sintácticos de la cabecera de fragmento de profundidad representativo de unas características del fragmento de profundidad de la cabecera del fragmento de referencia, y en el que la cabecera del fragmento de referencia es de un componente de vista de la misma unidad de

acceso que la vista actual.

SUMARIO

5 La presente invención está definida en las reivindicaciones adjuntas, a las cuales se hará referencia a continuación. En general, la presente divulgación describe unas técnicas para admitir el renderizado de vídeo tridimensional (3D). En particular, las técnicas de la presente divulgación se refieren a la codificación y decodificación de contenido de vídeo 3D. La presente divulgación también propone técnicas de señalización para unidades de bloques codificados de datos de vídeo. Por ejemplo, la presente divulgación propone reutilizar elementos sintácticos incluidos en una
10 cabecera de fragmento de componentes de vista de textura para componentes de vista de profundidad correspondientes, y viceversa. Además, la presente divulgación propone reutilizar elementos sintácticos en la información de cabecera, por ejemplo, información de cabecera de fragmento de componentes de vista de profundidad para componentes de vista de textura.

15 En un códec 3D, un componente de vista de cada vista de datos de vídeo en un instante de tiempo específico puede incluir un componente de vista de textura y un componente de vista de profundidad. El componente de vista de textura puede incluir componentes de luminancia (Y) y crominancia (Cb y Cr). Los componentes de luminancia (brillo) y crominancia (color) se denominan colectivamente en la presente memoria «componentes de textura». El componente de vista de profundidad puede ser de un mapa de profundidad de una imagen. En el renderizado de
20 imágenes 3D, los mapas de profundidad incluyen componentes de profundidad que son representativos de los valores de profundidad, por ejemplo, para componentes de textura correspondientes. Los componentes de vista de profundidad se pueden utilizar para generar vistas virtuales desde una perspectiva de visualización dada.

25 Los elementos sintácticos para componentes de profundidad y componentes de textura se pueden señalar con una unidad de bloque codificado. Las unidades de bloques codificados, denominadas también simplemente «bloques codificados» en la presente divulgación, pueden corresponder a macrobloques en UIT-T H.264/AVC (Codificación de vídeo avanzada) o unidades de codificación de Codificación de vídeo de alta eficiencia (HEVC).

30 En un aspecto, un procedimiento de codificación de datos de vídeo incluye codificar uno o más bloques de datos de vídeo representativos de la información de textura de al menos una parte de una trama de los datos de vídeo. El procedimiento incluye además procesar un fragmento de textura para un componente de vista de textura de una vista actual asociada con una unidad de acceso, comprendiendo el fragmento de textura el uno o más bloques codificados y una cabecera de fragmento de textura que comprende un conjunto de elementos sintácticos representativos de características del fragmento de textura. El procedimiento también incluye codificar información
35 de profundidad representativa de unos valores de profundidad para al menos la parte de la trama. El procedimiento incluye además procesar un fragmento de profundidad para un componente de vista de profundidad correspondiente al componente de vista de textura de la vista, comprendiendo el fragmento de profundidad la información de profundidad codificada y una cabecera de fragmento de profundidad que comprende un conjunto de elementos sintácticos representativos de características del fragmento de profundidad. Procesar el fragmento de textura o el
40 fragmento de profundidad puede incluir además predecir al menos un elemento sintáctico de al menos uno del conjunto de elementos sintácticos representativos de características del fragmento de textura o el conjunto de elementos sintácticos representativos de características del fragmento de profundidad de una cabecera de fragmento de referencia, respectivamente, en el que la cabecera de fragmento de referencia es de un componente de vista de la misma unidad de acceso.

45 En otro aspecto, la presente divulgación describe un dispositivo para codificar datos. El dispositivo incluye un codificador de vídeo configurado para codificar uno o más bloques de datos de vídeo representativos de información de textura de al menos una parte de una trama de los datos de vídeo, procesar un fragmento de textura para un componente de vista de textura de una vista actual asociada con una unidad de acceso, comprendiendo el
50 fragmento de textura el uno o más bloques codificados y una cabecera de fragmento de textura que comprende un conjunto de elementos sintácticos representativos de características del fragmento de textura, codificar información de profundidad representativa de valores de profundidad para al menos la parte de la trama y procesar un fragmento de profundidad para un componente de vista de profundidad correspondiente al componente de vista de textura de la vista, comprendiendo el fragmento de profundidad la información de profundidad codificada y una cabecera de
55 fragmento de profundidad que comprende un conjunto de elementos sintácticos representativos de características del fragmento de profundidad, en el que procesar el fragmento de textura o el fragmento de profundidad comprende predecir al menos un elemento sintáctico de al menos uno del conjunto de elementos sintácticos representativos de características del fragmento de textura o el conjunto de elementos sintácticos representativos de características del fragmento de profundidad de una cabecera de fragmento de referencia, respectivamente, en el que la cabecera de
60 fragmento de referencia es de un componente de vista de la misma unidad de acceso.

65 En otro aspecto más, la presente divulgación describe un producto de programa informático que puede estar configurado para ejecutar una o más de las técnicas descritas. El producto de programa informático incluye un medio de almacenamiento legible por ordenador que tiene almacenadas instrucciones que, al ejecutarse, hacen que un procesador de un dispositivo de codificación de vídeo codifique uno o más bloques de datos de vídeo representativos de información de textura de al menos una parte de una trama de los datos de vídeo. Las

instrucciones además hacen que el procesador procese un fragmento de textura para un componente de vista de textura de una vista actual asociada con una unidad de acceso, comprendiendo el fragmento de textura el uno o más bloques codificados y una cabecera de fragmento de textura que comprende un conjunto de elementos sintácticos representativos de características del fragmento de textura. Las instrucciones además hacen que el procesador codifique información de profundidad representativa de valores de profundidad para al menos la parte de la trama. Asimismo, las instrucciones hacen además que el procesador procese un fragmento de profundidad para un componente de vista de profundidad correspondiente al componente de vista de textura de la vista, comprendiendo el fragmento de profundidad la información de profundidad codificada y una cabecera de fragmento de profundidad que comprende un conjunto de elementos sintácticos representativos de características del fragmento de profundidad, en el que procesar el fragmento de textura o el fragmento de profundidad comprende predecir al menos un elemento sintáctico de al menos uno del conjunto de elementos sintácticos representativos de características del fragmento de textura o el conjunto de elementos sintácticos representativos de las características del fragmento de profundidad de una cabecera de fragmento de referencia, respectivamente, en el que la cabecera de fragmento de referencia es de un componente de vista de la misma unidad de acceso.

En otro aspecto, se describe un dispositivo para procesar datos de vídeo que incluye medios para codificar uno o más bloques de datos de vídeo representativos de información de textura de al menos una parte de una trama de los datos de vídeo. El dispositivo incluye además medios para procesar un fragmento de textura para un componente de vista de textura de una vista de corriente asociada con una unidad de acceso, comprendiendo el fragmento de textura el uno o más bloques codificados y una cabecera de fragmento de textura que comprende un conjunto de elementos sintácticos representativos de características del fragmento de textura y medios para codificar información de profundidad representativa de valores de profundidad para al menos la parte de la trama. El dispositivo incluye además medios para procesar un fragmento de profundidad para un componente de vista de profundidad correspondiente al componente de vista de textura de la vista, comprendiendo el fragmento de profundidad la información de profundidad codificada y una cabecera de fragmento de profundidad que comprende un conjunto de elementos sintácticos representativos de características del fragmento de profundidad, en el que los medios para procesar el fragmento de textura o el fragmento de profundidad comprenden medios para predecir al menos un elemento sintáctico de al menos uno del conjunto de elementos sintácticos representativos de características del fragmento de textura o el conjunto de elementos sintácticos representativos de características del fragmento de profundidad de una cabecera de fragmento de referencia, respectivamente, en el que la cabecera de fragmento de referencia es de un componente de vista de la misma unidad de acceso.

Las técnicas descritas en la presente divulgación pueden implementarse en hardware, software, firmware o cualquier combinación de estos. Si se implementa en software, el software puede ejecutarse en un procesador, que puede referirse a uno o más procesadores, tales como un microprocesador, un circuito integrado específico de la aplicación (ASIC), una matriz de puertas programables in situ (FPGA) o un procesador de señales digitales (DSP) u otro tipo de circuito lógico integrado o discreto equivalente. El software que comprende instrucciones para ejecutar las técnicas se puede almacenar inicialmente en un medio legible por ordenador y cargar y ejecutar mediante un procesador.

La presente divulgación también contempla medios legibles por ordenador que comprenden instrucciones para hacer que un procesador realice cualquiera de una variedad de técnicas descritas en la presente divulgación. En algunos casos, el medio legible por ordenador puede formar parte de un producto de programa informático, que puede ponerse a la venta para los fabricantes y/o utilizarse en un dispositivo. El producto de programa informático puede incluir el medio legible por ordenador y, en algunos casos, también puede incluir materiales de embalaje.

Los detalles de uno o más aspectos de la divulgación se exponen en los dibujos adjuntos y la descripción siguiente. Otras características, objetivos y ventajas de las técnicas descritas en la presente divulgación resultarán evidentes a partir de la descripción y los dibujos, y a partir de las reivindicaciones.

50 BREVE DESCRIPCIÓN DE LOS DIBUJOS

La FIG. 1 es un diagrama de bloques que ilustra un ejemplo de sistema de codificación y decodificación de vídeo, de acuerdo con las técnicas de la presente divulgación.

La FIG. 2 es un diagrama de bloques que ilustra con más detalle un ejemplo del codificador de vídeo de la FIG. 1, de acuerdo con las técnicas de la presente divulgación.

La FIG. 3 es un diagrama conceptual que ilustra un ejemplo de orden de decodificación multivista, de acuerdo con las técnicas de la presente divulgación.

La FIG. 4 es un diagrama de un ejemplo de estructura de predicción MVC para codificación de vídeo multivista, de acuerdo con las técnicas de la presente divulgación.

La FIG. 5 es un diagrama de bloques que ilustra con más detalle un ejemplo del decodificador de la FIG. 1, de acuerdo con las técnicas de la presente divulgación.

La FIG. 6 es un diagrama de flujo que ilustra un ejemplo de operación de un codificador de vídeo, de acuerdo con las técnicas de la presente divulgación.

DESCRIPCIÓN DETALLADA

5 La presente divulgación describe técnicas que un codificador puede aplicar y un decodificador puede utilizar durante al menos la fase de predicción inter de al menos un proceso de codificación o decodificación de vídeo. Las técnicas descritas están relacionadas con la codificación de contenido de vídeo tridimensional («3D»). El contenido de vídeo 3D puede representarse, por ejemplo, como bloques codificados de vídeo multivista más profundidad («MVD»). Es
10 decir, estas técnicas se pueden aplicar para codificar o decodificar un flujo de bits que se asemeja a un flujo de bits de codificación de vídeo multivista (MVC), donde cualquiera o todas las vistas del flujo de bits de MVC pueden incluir además información de profundidad. Como se describe en el presente documento, la señalización tiene lugar en un flujo de bits codificado.

15 Más específicamente, algunas técnicas de acuerdo con la presente divulgación implican la recepción de al menos una imagen bidimensional que tiene componentes de vista de textura y componentes de vista de profundidad. Algunos componentes de vista de textura y componentes de vista de profundidad pueden codificarse juntos como un solo bloque codificado o como bloques separados. Una imagen puede dividirse en fragmentos de imágenes. Los
20 elementos sintácticos para codificar componentes de vista de textura se pueden señalar en una cabecera de fragmento. Algunos elementos sintácticos para los componentes de vista de profundidad se pueden predecir a partir de los elementos sintácticos para los componentes de vista de textura correspondientes a los componentes de vista de profundidad. Las técnicas de la presente divulgación se refieren a la codificación, decodificación y señalización de datos utilizados para renderizar datos de vídeo tridimensional a partir de datos de vídeo bidimensional, basándose en datos de mapa de profundidad estimados para los datos de vídeo bidimensional. En algunos ejemplos, los
25 componentes de vista de textura se codifican mediante técnicas diferentes a las utilizadas para codificar la información de profundidad. En la presente divulgación, el término «codificación» puede referirse a cualquiera de los procesos de codificación y decodificación o a ambos.

La conversión de vídeo basada en la estimación de profundidad y la síntesis de vistas virtuales se utiliza para crear
30 imágenes 3D, por ejemplo, para aplicaciones de vídeo 3D. En particular, las vistas virtuales de una escena se pueden utilizar para crear una vista 3D de la escena. La generación de una vista virtual de una escena basada en una vista existente de la escena se consigue convencionalmente estimando valores de profundidad de objeto antes de sintetizar la vista virtual. La estimación de profundidad es un proceso de estimación de distancias absolutas o relativas entre objetos y un plano de cámara a partir de pares estéreo o contenido monoscópico. Tal como se utiliza
35 en el presente documento, la información de profundidad incluye información útil en el procesamiento de vídeo tridimensional, tal como un mapa de profundidad (por ejemplo, valores de profundidad para cada píxel) o un mapa de paralaje (por ejemplo, disparidad horizontal para cada píxel).

La información de profundidad estimada, normalmente representada por un mapa de profundidad de imagen en
40 escala de grises, puede utilizarse para generar un ángulo arbitrario para vistas virtuales mediante técnicas de renderizado basadas en imagen de profundidad (DIBR). En comparación con los sistemas tradicionales de televisión tridimensional (3DTV) en los que las secuencias multivista afrontan los retos de la compresión inter de vistas eficiente, un sistema basado en mapa de profundidad puede reducir el uso del ancho de banda transmitiendo solo una o varias vistas junto con el (los) mapa(s) de profundidad, que pueden codificarse eficazmente. Los mapas de
45 profundidad utilizados en la conversión basada en mapas de profundidad pueden ser controlables (por ejemplo, a través de escalado) por los usuarios finales antes de que se utilicen los mapas de profundidad en la síntesis de vistas. Pueden generarse vistas virtuales personalizadas con diferentes cantidades de profundidad percibidas. También se puede realizar una estimación de la profundidad mediante vídeo monoscópico en el que solamente está disponible un contenido de visita 2D.

50 Algunas técnicas de la presente divulgación se describirán con referencia a la extensión de codificación de vídeo multivista (MVC) de la norma H.264/AVC (codificación de vídeo avanzada). El último borrador conjunto de MVC se describe en el documento de JVT-AD007, «Editors' draft revision to UIT-T Rec. H.264 | ISO/IEC 14496-10 Advanced Video Coding», 30th JVT meeting, Ginebra, Suiza, enero-febrero de 2009, disponible en http://wftp3.itu.int/av-arch/jvt-site/2009_01_Geneva/JVT-AD007. Si bien la siguiente descripción se realizará en términos de la norma H.264/AVC, debe entenderse que las técnicas de la presente divulgación pueden ser aplicables para su uso con
55 otros procesos de codificación de vídeo multivista, o con futuras normas de codificación de vídeo multivista o propuestas actualmente.

60 En H.264/AVC, cada unidad de capa de abstracción de red (NAL) tiene una cabecera de fragmento si contiene datos de vídeo codificados. La cabecera de fragmento puede contener la siguiente información: dirección del fragmento, ubicación del macrobloque de inicio, el tipo de fragmento, el conjunto de parámetros de imagen (PPS) que se va a utilizar y el parámetro de cuantificación delta (QP) entre el QP inicial del fragmento y el QP señalado en el PPS, el orden de las imágenes de referencia (frame_num) y el orden de visualización de la imagen actual (por ejemplo, el
65 recuento de orden de imágenes (POC)), elementos sintácticos relacionados con la construcción de la lista de imágenes de referencia, elementos sintácticos relacionados con la operación de gestión de la memoria y elementos

sintácticos relacionados con la predicción ponderada.

Las técnicas descritas en el presente documento pueden aplicarse para predecir elementos sintácticos para un componente de vista de profundidad a partir de elementos sintácticos almacenados en una cabecera de fragmento para componentes de vista de textura coubicados de la misma vista. Por ejemplo, se pueden incluir valores de los elementos sintácticos que son comunes al fragmento de profundidad y el fragmento de textura en la cabecera de fragmento para los componentes de vista de textura, pero no en el fragmento para los componentes de vista de profundidad asociados. Es decir, un codificador o decodificador de vídeo puede codificar elementos sintácticos que son comunes al fragmento de profundidad y el fragmento de textura en la cabecera de fragmento para los componentes de vista de textura que no están presentes en la cabecera de fragmento para los componentes de vista de profundidad. Por ejemplo, se puede definir un primer valor para un primer elemento sintáctico de la cabecera de fragmento para los componentes de vista de textura. La cabecera de fragmento para los componentes de vista de profundidad también comparte el primer elemento sintáctico, es decir, el primer elemento sintáctico es común tanto a la cabecera de fragmento de textura como a la cabecera de fragmento de profundidad. El primer elemento sintáctico para los componentes de vista de profundidad tiene un segundo valor. Sin embargo, la cabecera de fragmento para el componente de vista de profundidad no incluye el primer elemento sintáctico. De acuerdo con las técnicas descritas en el presente documento, el segundo valor del primer elemento sintáctico puede predecirse a partir del primer valor.

La presente divulgación presenta las siguientes técnicas para la predicción de cabecera de fragmento. De acuerdo con un ejemplo de la divulgación, la predicción se controla en el nivel de cabecera de fragmento, y la cabecera de fragmento de referencia puede definirse como un componente de vista en la misma vista de la misma unidad de acceso que la vista actual o una textura de vista básica compatible con la AVC de la misma unidad de acceso que la vista actual.

En otro ejemplo, la cabecera de fragmento de referencia de un componente de vista de profundidad puede ser la cabecera de fragmento del componente de vista de textura de la misma vista o la cabecera de fragmento del componente de vista de profundidad de la primera vista según el orden de decodificación. En otro ejemplo, se puede señalar un id de índice de orden de vista delta para indicar a partir de qué vista de la misma unidad de acceso se predice la cabecera de fragmento.

En algunos ejemplos, se puede señalar un indicador para indicar si la cabecera de fragmento de referencia es del componente de vista de profundidad o del componente de vista de textura de una vista específica. En otro ejemplo, los elementos sintácticos de la extensión de la cabecera de fragmento, si están presentes en la cabecera de fragmento, pueden reordenarse de una manera que es diferente a la de los elementos sintácticos de la cabecera de fragmento en la cabecera de fragmento de AVC.

En otro ejemplo, los elementos sintácticos `slice_type`, `num_ref_idx_l0_active_minus1`, `num_ref_idx_l1_active_minus1` y la modificación de lista de imágenes de referencia pueden predecirse selectivamente. Es decir, pueden predecirse o no basándose en un indicador de control situado en cada extensión 3DV de cabecera de fragmento. Cuando se habilita la predicción de la modificación de lista de imágenes de referencia, puede ser posible refinar la modificación de lista de imágenes de referencia basándose en una tabla de sintaxis de exclusión o inserción de lista de imágenes de referencia recién introducida.

En otro ejemplo de la divulgación, después de que se indique un componente de vista de profundidad o textura, se puede señalar además un ID de fragmento del fragmento de un componente de vista dado de una unidad de acceso. En algunos ejemplos, la unidad de acceso que comprende el fragmento del componente de vista dado puede ser la misma unidad de acceso correspondiente al componente de vista de profundidad o textura.

En otro ejemplo de la divulgación, se puede señalar un ID de fragmento para cada fragmento de cada componente de vista. El ID de fragmento puede señalarse en la cabecera de fragmento o en la cabecera de unidad NAL. Además, las indicaciones anteriores pueden señalarse en la extensión 3DV de la cabecera.

Las técnicas de la divulgación pueden utilizar codificación inter basada en bloques. La codificación inter basada en bloques es una técnica de codificación que se basa en la predicción temporal para reducir o eliminar la redundancia temporal entre bloques de vídeo de unidades codificadas sucesivas de una secuencia de vídeo. Las unidades codificadas pueden comprender tramas de vídeo, fragmentos de tramas de vídeo, grupos de imágenes u otra unidad definida de bloques de vídeo codificados. Para la codificación inter, un codificador de vídeo realiza estimación de movimiento y compensación de movimiento para estimar el movimiento entre bloques de vídeo de dos o más unidades codificadas adyacentes. Mediante técnicas para estimación de movimiento, el codificador de vídeo genera vectores de movimiento, que indican un desplazamiento de bloques de vídeo en relación con los correspondientes bloques de vídeo de predicción de una o más tramas de referencia u otras unidades codificadas. Mediante técnicas para compensación de movimiento, el codificador de vídeo utiliza los vectores de movimiento para generar bloques de vídeo de predicción a partir de la una o más tramas de referencia u otras unidades codificadas. Después de la compensación de movimiento, el codificador de vídeo calcula los bloques de vídeo residuales restando bloques de vídeo de predicción de los bloques de vídeo originales que se están codificando.

Los componentes de vista de referencia (RVC) pueden incluir varios fragmentos de textura o profundidad. En algunos ejemplos, cuando los componentes de vista de referencia comprenden varios fragmentos, se puede utilizar un fragmento coubicado al determinar los elementos sintácticos de un fragmento actual. De forma alternativa, se puede utilizar un primer fragmento del RVC para determinar elementos sintácticos del fragmento actual. En otros ejemplos, se puede utilizar otro fragmento del RVC para determinar elementos sintácticos comunes del fragmento actual.

La FIG. 1 es un diagrama de bloques que ilustra un ejemplo de sistema de codificación y decodificación de vídeo de acuerdo con las técnicas de la presente divulgación. Como se ilustra en el ejemplo de la FIG. 1, el sistema incluye un dispositivo de origen 12 que transmite vídeo codificado a un dispositivo de destino 16 a través de un enlace 15. El enlace 15 puede comprender cualquier tipo de medio o dispositivo capaz de desplazar los datos de vídeo codificado desde el dispositivo de origen 12 al dispositivo de destino 16. En un ejemplo, el enlace 15 comprende un medio de comunicación para permitir al dispositivo de origen 12 transmitir datos de vídeo codificado directamente al dispositivo de destino 16 en tiempo real. Los datos de vídeo codificado pueden modularse de acuerdo con una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitirse al dispositivo de destino 16. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrico o cableado, tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión física. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área extensa o una red global tal como Internet. El medio de comunicación puede incluir routers, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo de origen 12 al dispositivo de destino 16.

El dispositivo de origen 12 y el dispositivo de destino 16 pueden comprender cualquiera de una amplia gama de dispositivos. En algunos casos, cualquiera de entre el dispositivo de origen 12 y el dispositivo de destino 16 o ambos pueden comprender dispositivos de comunicación inalámbricos, tales como teléfonos inalámbricos, los denominados radiotelefonos celulares o por satélite, o cualquier dispositivo inalámbrico que pueda comunicar información de vídeo a través de un enlace 15, en cuyo caso el enlace 15 es inalámbrico. Sin embargo, las técnicas de la presente divulgación, que se refieren a la codificación de bloques de datos de vídeo que incluyen tanto información de textura como de profundidad, no están necesariamente limitadas a aplicaciones o configuraciones inalámbricas. Las técnicas también pueden ser útiles en una amplia gama de configuraciones y dispositivos diferentes, incluidos los dispositivos que se comunican a través de cables físicos, fibras ópticas u otros medios físicos o inalámbricos. Además, las técnicas de codificación o decodificación también pueden aplicarse en un dispositivo autónomo que no necesariamente se comunica con otro dispositivo. Por ejemplo, el decodificador de vídeo 28 puede residir en un reproductor de medios digitales u otro dispositivo, y recibir datos de vídeo codificados a través de medios de transmisión en continuo, descarga o almacenamiento. Por lo tanto, la descripción del dispositivo de origen 12 y el dispositivo de destino 16 que se comunican entre sí se realiza con fines de ilustración de un ejemplo de implementación, y no debe considerarse limitativa de las técnicas descritas en la presente divulgación, que pueden aplicarse a la codificación de vídeo en general en una variedad de entornos, aplicaciones o implementaciones.

De forma alternativa, los datos codificados pueden transmitirse desde la interfaz de salida 24 hasta un dispositivo de almacenamiento 17. De forma similar, se puede acceder a los datos codificados desde el dispositivo de almacenamiento 17 mediante el codificador de vídeo 28. El dispositivo de almacenamiento 17 puede incluir cualquiera de entre una diversidad de medios de almacenamiento de datos de acceso distribuido o local, tales como una unidad de disco fijo, discos Blu-ray, DVD, CD-ROM, memoria flash, memoria volátil o no volátil o cualquier otro medio de almacenamiento digital adecuado para almacenar datos de vídeo codificado. En un ejemplo adicional, el dispositivo de almacenamiento 17 puede corresponder a un servidor de archivos o a otro dispositivo de almacenamiento intermedio que pueda retener el vídeo codificado generado por el dispositivo de origen 12. El dispositivo de destino 16 puede acceder a los datos de vídeo almacenados desde el dispositivo de almacenamiento 17 a través de transmisión en continuo o descarga. El servidor de archivos puede ser cualquier tipo de servidor capaz de almacenar datos de vídeo codificado y transmitir esos datos de vídeo codificado al dispositivo de destino 16. Entre los ejemplos de servidores de archivos se incluyen un servidor web (por ejemplo, para un sitio web), un servidor FTP, dispositivos de almacenamiento conectados a la red (NAS) o una unidad de disco local. El dispositivo de destino 16 puede acceder a los datos de vídeo codificado a través de cualquier conexión de datos estándar, incluida una conexión a Internet. Esta puede incluir un canal inalámbrico (por ejemplo, una conexión wifi), una conexión por cable (por ejemplo, DSL, módem de cable, etc.), o una combinación de ambos que sea adecuada para acceder a datos de vídeo codificado almacenados en un servidor de archivos. La transmisión de datos de vídeo codificado desde el dispositivo de almacenamiento 17 puede ser una transmisión en continuo, una transmisión de descarga o una combinación de ambas.

En el ejemplo de la FIG. 1, el dispositivo de origen 12 incluye una fuente de vídeo 20, una unidad de procesamiento de profundidad 21, un codificador de vídeo 22 y una interfaz de salida 24. El dispositivo de destino 16 incluye una interfaz de entrada 26, un decodificador de vídeo 28 y un dispositivo de visualización 30. De acuerdo con la presente divulgación, el codificador de vídeo 22 del dispositivo de origen 12 puede estar configurado para aplicar una o más de las técnicas de la presente divulgación como parte de un proceso de codificación de vídeo. De manera similar, el decodificador de vídeo 28 del dispositivo de destino 16 puede estar configurado para aplicar una o más de las

técnicas de la presente divulgación como parte de un proceso de decodificación de vídeo.

El codificador de vídeo 22 también puede aplicar procedimientos de codificación de transformada, cuantificación y entropía para reducir aún más la velocidad de bits asociada con la transmisión de bloques residuales. Las técnicas de transformada pueden comprender transformadas de coseno discretas (DCT) o procesos conceptualmente similares. De forma alternativa, pueden utilizarse transformadas de ondícula, transformadas enteras u otros tipos de transformadas. A título de ejemplo, en un proceso de DCT, un conjunto de valores de píxel se convierte en coeficientes de transformada, que representan la energía de los valores de píxel en el dominio de la frecuencia. El codificador de vídeo 22 puede también cuantificar los coeficientes de transformada, que en general pueden implicar un proceso que reduce el número de bits asociados con el coeficiente de transformada correspondiente. La codificación de entropía puede incluir uno o más procesos que comprimen colectivamente datos para transmitir a un flujo de bits, donde los datos comprimidos pueden incluir, por ejemplo, una secuencia de modalidades de codificación, información de movimiento, patrones de bloques codificados y coeficientes de transformada cuantificados. Entre los ejemplos de codificación de entropía se incluyen, pero no se limitan a, la codificación de longitud variable adaptativa al contexto (CAVLC) y la codificación aritmética binaria adaptativa al contexto (CABAC).

Un bloque de vídeo codificado puede representarse mediante información de predicción que se puede utilizar para crear o identificar un bloque predictivo, y un bloque residual de datos que se puede aplicar al bloque predictivo para recrear el bloque original. La información de predicción puede comprender uno o más vectores de movimiento que se utilizan para identificar el bloque de datos predictivo. Mediante los vectores de movimiento, el decodificador de vídeo 28 puede ser capaz de reconstruir los bloques predictivos que se utilizaron para codificar los bloques residuales. Por lo tanto, dado un conjunto de bloques residuales y un conjunto de vectores de movimiento (y posiblemente sintaxis adicional), el decodificador de vídeo 28 puede reconstruir una trama de vídeo que se codificó originalmente. La codificación inter basada en estimación de movimiento y compensación de movimiento puede alcanzar cantidades relativamente altas de compresión sin pérdida de datos excesiva, porque las tramas de vídeo sucesivas u otros tipos de unidades codificadas a menudo son similares. Una secuencia de vídeo codificada puede comprender bloques de datos residuales, vectores de movimiento (cuando se ha sometido a codificación de predicción inter), indicaciones de modalidades de predicción intra para predicción intra, y elementos sintácticos.

El codificador de vídeo 22 también puede utilizar técnicas de predicción intra para codificar bloques de vídeo en relación con bloques de vídeo contiguos de una trama o un fragmento común. De esta manera, el codificador de vídeo 22 predice los bloques espacialmente. El codificador de vídeo 22 puede configurarse con una variedad de modalidades de predicción intra, que en general corresponden a varias direcciones de predicción espaciales. Al igual que con la estimación de movimiento, el codificador de vídeo 22 puede configurarse para seleccionar una modalidad de predicción intra basándose en un componente de luminancia de un bloque y reutilizar, a continuación, la modalidad de predicción intra para codificar componentes de crominancia del bloque. Además, de acuerdo con las técnicas de la presente divulgación, el codificador de vídeo 22 puede reutilizar la modalidad de predicción intra para codificar un componente de profundidad del bloque.

Mediante la reutilización de información de modalidad de movimiento y predicción intra para codificar un componente de profundidad de un bloque, estas técnicas pueden simplificar el proceso de codificación de mapas de profundidad. Además, las técnicas descritas en el presente documento pueden mejorar la eficiencia del flujo de bits. Es decir, el flujo de bits solo necesita indicar algunos elementos sintácticos una vez en una cabecera de fragmento para un componente de vista de textura, en lugar de señalar el elemento sintáctico adicional en una cabecera de fragmento para un fragmento de componentes de vista de profundidad.

Opcionalmente, un componente de vista de textura también puede reutilizar su componente de vista de profundidad correspondiente de la misma manera.

También aquí, el sistema ilustrado 10 de la FIG. 1 es simplemente un ejemplo. Las diversas técnicas de la presente divulgación pueden realizarse mediante cualquier dispositivo de codificación que admita la codificación predictiva basada en bloques, o mediante cualquier dispositivo de decodificación que admita la decodificación predictiva basada en bloques. El dispositivo de origen 12 y el dispositivo de destino 16 son simplemente ejemplos de dichos dispositivos de codificación, en los que el dispositivo de origen 12 genera datos de vídeo codificados para su transmisión al dispositivo de destino 16. En algunos casos, los dispositivos 12 y 16 pueden funcionar de manera esencialmente simétrica, de modo que cada uno de los dispositivos 12 y 16 incluye componentes de codificación y de decodificación de vídeo. Por lo tanto, el sistema 10 puede admitir una transmisión de vídeo unidireccional o bidireccional entre los dispositivos de vídeo 12 y 16, por ejemplo, para transmisión en continuo de vídeo, reproducción de vídeo, radiodifusión de vídeo o videotelefonía.

La fuente de vídeo 20 del dispositivo de origen 12 incluye un dispositivo de captación de vídeo, tal como una cámara de vídeo, un archivo de vídeo que contiene vídeo captado previamente o vídeo procedente de un proveedor de contenido de vídeo. De forma alternativa, la fuente de vídeo 20 puede generar datos basados en gráficos de ordenador como vídeo de origen, o una combinación de vídeo en directo, vídeo archivado y/o vídeo generado por ordenador. En algunos casos, si la fuente de vídeo 20 es una videocámara, el dispositivo de origen 12 y el dispositivo de destino 16 pueden formar los denominados teléfonos con cámara o videoteléfonos u otros dispositivos

móviles configurados para manipular datos de vídeo, tales como dispositivos informáticos de tableta. En cada caso, el vídeo captado, precaptado o generado por ordenador puede codificarse mediante el codificador de vídeo 22. La fuente de vídeo 20 capta una vista y la envía a la unidad de procesamiento de profundidad 21.

- 5 La fuente de vídeo 20 envía la vista 2 a la unidad de procesamiento de profundidad 21 para calcular la imagen de profundidad de los objetos de la vista 2. En algunos ejemplos, la vista 2 comprende más de una vista. Se determina una imagen de profundidad para los objetos de la vista 2 captados por la fuente de vídeo 20. La unidad de procesamiento de profundidad 21 está configurada para calcular automáticamente los valores de profundidad para los objetos de la imagen de la vista 2. Por ejemplo, la unidad de procesamiento de profundidad 21 calcula valores de
- 10 profundidad para objetos, basándose en información de luminancia. En algunos ejemplos, la unidad de procesamiento de profundidad 21 está configurada para recibir información de profundidad desde un usuario. En algunos ejemplos, la fuente de vídeo 20 capta dos vistas de una escena desde perspectivas diferentes y, a continuación, calcula información de profundidad para objetos de la escena basándose en la disparidad entre los objetos de las dos vistas. En varios ejemplos, la fuente de vídeo 20 comprende una cámara bidimensional estándar,
- 15 un sistema de dos cámaras que proporciona una vista estereoscópica de una escena, un conjunto de cámaras que capta varias vistas de la escena o una cámara que capta una vista más información de profundidad.

La unidad de procesamiento de profundidad 21 envía componentes de vista de textura 4 y componentes de vista de profundidad 6 al codificador de vídeo 22. La unidad de procesamiento de profundidad 21 también puede enviar la vista 2 directamente al codificador de vídeo 22. La información de profundidad 6 comprende una imagen de mapa de profundidad para la vista 2. Una imagen de mapa de profundidad puede comprender un mapa de valores de profundidad para cada zona de píxeles asociados con un área (por ejemplo, un bloque, un fragmento o una trama) que se va a visualizar. Una zona de píxeles incluye un único píxel o un grupo de uno o más píxeles. Algunos ejemplos de mapas de profundidad tienen un componente de profundidad por píxel. En otros ejemplos, hay varios

20 componentes de profundidad por píxel. Los mapas de profundidad pueden codificarse de una manera sustancialmente similar a los datos de textura, por ejemplo, mediante predicción intra o predicción inter en relación con otros datos de profundidad codificados previamente. En otros ejemplos, los mapas de profundidad se codifican de una manera diferente a los datos de textura.

30 El mapa de profundidad puede estimarse en algunos ejemplos. Cuando hay más de una vista, se puede utilizar la correspondencia estéreo para estimar mapas de profundidad. Sin embargo, en la conversión 2D a 3D, la estimación de la profundidad puede ser más difícil. No obstante, el mapa de profundidad estimado mediante varios procedimientos puede utilizarse para el renderizado 3D basado en el renderizado basado en imagen de profundidad (DIBR).

35 Aunque la fuente de vídeo 20 puede incluir varias vistas de escena y la unidad de procesamiento de profundidad 21 puede calcular información de profundidad basándose en las diversas vistas, el dispositivo de origen 12 puede transmitir en general una vista más información de profundidad para cada vista de una escena.

40 Cuando la vista 2 es una imagen digital fija, el codificador de vídeo 22 puede configurarse para codificar la vista 2 como, por ejemplo, una imagen del Grupo conjunto de expertos en fotografía (JPEG). Cuando la vista 2 es una trama de datos de vídeo, el codificador de vídeo 22 está configurado para codificar la primera vista 50 de acuerdo con una norma de codificación de vídeo, tal como, por ejemplo, las del Grupo de expertos de imágenes en movimiento (MPEG), la Organización internacional de normalización (ISO)/Comisión electrotécnica internacional

45 (IEC) MPEG-1 Visual, la ISO/IEC MPEG-2 Visual, la Unión internacional de telecomunicaciones (UIT) H.261, UIT-T H.262, UIT-T H.263, UIT-T H.264/MPEG-4, la codificación de vídeo avanzada (AVC) H.264, la próxima norma de codificación de vídeo de alta eficiencia (HEVC) (también conocida como H.265) u otras normas de codificación de vídeo. El codificador de vídeo 22 puede incluir información de profundidad 6 junto con la imagen codificada para formar el bloque codificado 8, que incluye datos de imagen codificados junto con información de profundidad 6. El codificador de vídeo 22 pasa el bloque codificado 8 a la interfaz de salida 24. El bloque codificado 8 puede transferirse a la interfaz de entrada 26 en un flujo de bits que incluye información de señalización junto con el bloque codificado 8, a través del enlace 15.

55 La información de vídeo codificada incluye componentes de textura 4 e información de profundidad 6. Los componentes de textura 4 pueden incluir componentes de luminancia (luma) y crominancia (croma) de la información de vídeo. Los componentes de luma en general describen el brillo, mientras que los componentes de crominancia en general describen tonalidades de color. La unidad de procesamiento de profundidad 21 extrae información de profundidad 6 de un mapa de profundidad de la vista 2. El codificador de vídeo 22 puede codificar componentes de vista de textura 4 y componentes de vista de profundidad 6 en un único bloque codificado 8 de datos de vídeo codificados. Del mismo modo, el codificador de vídeo 22 puede codificar el bloque, de modo que la información de modalidad de movimiento o predicción intra para el componente de luma se reutilice para los componentes de croma y el componente de profundidad. Los elementos sintácticos utilizados para los componentes de vista de textura pueden utilizarse para predecir elementos sintácticos similares para los componentes de vista de profundidad.

65 En algunos ejemplos, el componente de vista de mapa de profundidad no se puede codificar mediante técnicas de predicción inter de vistas incluso cuando el componente de vista de textura correspondiente se codifica mediante

técnicas de predicción inter de vistas. Por ejemplo, el componente de vista de mapa de profundidad se puede predecir mediante predicción inter de vistas cuando el componente de vista de textura correspondiente se predice mediante predicción inter de vistas. Por ejemplo, la predicción inter de vistas de un componente de vista de textura predice la información de vista de textura a partir de datos de una vista diferente a la vista correspondiente al componente de vista de textura. Por el contrario, la información de vista de profundidad de predicción intra de vistas predice la información de profundidad a partir de datos de la misma vista que la vista correspondiente a la información de vista de profundidad.

A pesar de utilizar diferentes técnicas de predicción, algunos elementos sintácticos para el componente de vista de mapa de profundidad pueden predecirse a partir de los elementos sintácticos correspondientes de la cabecera de fragmento del componente de vista de textura correspondiente, o viceversa. Sin embargo, la información de cabecera de fragmento para el componente de vista de mapa de profundidad puede contener información relacionada con la construcción de una lista de imágenes de referencia. Es decir, la información relacionada con la construcción de lista de imágenes de referencia se puede señalar en la cabecera de fragmento para el componente de vista de mapa de profundidad. Por ejemplo, un número de imágenes de referencia que se utilizan y una indicación de qué imágenes de referencia se utilizan para predecir el componente de vista de mapa de profundidad se pueden señalar en la cabecera de fragmento para el componente de vista de mapa de profundidad. También se puede señalar una información similar en una cabecera de fragmento para el componente de vista de textura correspondiente.

En algunos ejemplos, el dispositivo de origen 12 incluye un módem que modula el bloque codificado 8 de acuerdo con una norma de comunicación, tal como el acceso múltiple por división de código (CDMA) u otra norma de comunicación. El módem puede incluir varios mezcladores, filtros, amplificadores u otros componentes diseñados para la modulación de señales. La interfaz de salida 24 puede incluir circuitos diseñados para transmitir datos, que incluyen amplificadores, filtros y una o más antenas. El bloque codificado 8 se transmite al dispositivo de destino 16 a través de la interfaz de salida 24 y el enlace 15. En algunos ejemplos, en lugar de transmitir a través de un canal de comunicación, el dispositivo de origen 12 almacena datos de vídeo codificados, incluidos bloques que tienen componentes de textura y profundidad, en un dispositivo de almacenamiento 32, tal como un disco de vídeo digital (DVD), un disco Blu-ray, una unidad flash o similar.

La interfaz de entrada 26 del dispositivo de destino 16 recibe información a través del enlace 15. En algunos ejemplos, el dispositivo de destino 16 incluye un módem que desmodula la información. Como la interfaz de salida 24, la interfaz de entrada 26 puede incluir circuitos diseñados para recibir datos, que incluyen amplificadores, filtros y una o más antenas. En algunos casos, la interfaz de salida 24 y/o la interfaz de entrada 26 pueden incorporarse dentro de un único componente de transceptor que incluye tanto circuitos de recepción como de transmisión. Un módem puede incluir varios mezcladores, filtros, amplificadores u otros componentes diseñados para la desmodulación de señales. En algunos casos, un módem puede incluir componentes para realizar la modulación y la desmodulación.

De nuevo, el proceso de codificación de vídeo realizado por el codificador de vídeo 22 puede implementar una o más de las técnicas descritas en el presente documento durante la codificación de predicción inter, que puede incluir estimación de movimiento y compensación de movimiento, y la codificación de predicción intra. El proceso de decodificación de vídeo realizado por el decodificador de vídeo 28 puede realizar también dichas técnicas durante una fase de compensación de movimiento del proceso de decodificación.

El término «codificador» se utiliza en el presente documento para referirse a un dispositivo o aparato informático especializado que realiza la codificación o la decodificación de vídeo. El término «codificador» se refiere en general a cualquier codificador de vídeo, decodificador de vídeo o codificador/decodificador combinado (códec). El término «codificación» se refiere a la codificación o la decodificación. El término «bloque codificado», «unidad de bloque codificado» o «unidad codificada» puede referirse a cualquier unidad de una trama de vídeo que puede decodificarse de manera independiente, tal como una trama completa, un fragmento de una trama, un bloque de datos de vídeo u otra unidad que pueda decodificarse de manera independiente definida de acuerdo con las técnicas de codificación utilizadas.

El dispositivo de visualización 30 muestra los datos de vídeo decodificados a un usuario, y puede comprender cualquiera de entre una variedad de uno o más dispositivos de visualización, tales como un tubo de rayos catódicos (CRT), una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de visualización. En algunos ejemplos, el dispositivo de visualización 30 corresponde a un dispositivo con capacidad de reproducción tridimensional. Por ejemplo, el dispositivo de visualización 30 puede comprender una pantalla estereoscópica, que el observador utiliza provisto de unas gafas. Las gafas pueden comprender gafas activas, en cuyo caso el dispositivo de visualización 30 alterna rápidamente entre imágenes de vistas diferentes de forma sincrónica y alterna la obturación de las lentes de las gafas activas. De forma alternativa, las gafas pueden comprender gafas pasivas, en cuyo caso el dispositivo de visualización 30 muestra imágenes de diferentes vistas simultáneamente, y las gafas pasivas pueden incluir lentes polarizadas que en general están polarizadas en direcciones ortogonales para filtrar entre las diferentes vistas.

- En el ejemplo de la FIG. 1, el enlace 15 puede comprender cualquier medio de comunicación inalámbrico o cableado, tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión física, o cualquier combinación de medios inalámbricos y cableados. El enlace 15 puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área extensa o una red global tal como Internet. El enlace 15 representa en general cualquier medio de comunicación adecuado o un conjunto de diferentes medios de comunicación, para transmitir datos de vídeo desde el dispositivo de origen 12 hasta el dispositivo de destino 16. El enlace 15 puede incluir routers, 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 16.
- El codificador de vídeo 22 y el decodificador de vídeo 28 pueden funcionar de acuerdo con una norma de compresión de vídeo, tal como la norma UIT-T H.264, de forma alternativa descrita como MPEG-4, Parte 10, Codificación avanzada de vídeo (AVC). Otras normas de compresión de vídeo, que están basadas en la norma UIT H.264/AVC y que pueden ser utilizadas por el codificador de vídeo 22 y el decodificador de vídeo 28, incluyen la norma de codificación de vídeo escalable (SVC), que es una extensión escalable para la norma UIT H.264/AVC.
- Otra norma de acuerdo con la cual el codificador de vídeo 22 y el decodificador de vídeo 28 pueden operar incluye la norma de codificación de vídeo multivista (MVC), que es una extensión multivista de la norma UIT H.264/AVC. Sin embargo, las técnicas de la presente divulgación no están limitadas a ninguna norma de codificación de vídeo particular.
- En algunos aspectos, cada uno de entre el codificador de vídeo 22 y el decodificador de vídeo 28 puede estar integrado con un codificador y decodificador de audio, y puede incluir unidades MUX-DEMUX adecuadas, u otro tipo de hardware y software, para gestionar la codificación tanto de audio como de vídeo en un flujo de datos común o en flujos de datos separados. Si procede, las unidades MUX-DEMUX pueden ajustarse al protocolo de multiplexado UIT H.223 o a otros protocolos, tales como el protocolo de datagramas de usuario (UDP).
- El codificador de vídeo 22 y el decodificador de vídeo 28 pueden implementarse como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), matrices de puertas programables in situ (FPGA), lógica discreta, software, hardware, firmware o cualquier combinación de estos. Cuando cualquiera o todas las técnicas de la presente divulgación se implementan en software, un dispositivo de implementación puede incluir además hardware para almacenar y/o ejecutar instrucciones para el software, por ejemplo, una memoria para almacenar las instrucciones y una o más unidades de procesamiento para ejecutar la instrucciones. El codificador de vídeo 22 y el decodificador de vídeo 28 pueden incluirse cada uno en uno o más codificadores o decodificadores, donde cualquiera de los cuales puede estar integrado como parte de un códec combinado que incluye capacidades de codificación y decodificación en un respectivo dispositivo móvil, dispositivo de abonado, dispositivo de radiodifusión, servidor o similares.
- Una secuencia de vídeo incluye típicamente una serie de tramas de vídeo, también denominadas imágenes de vídeo. Un codificador de vídeo 22 actúa en bloques de vídeo de tramas de vídeo individuales con el fin de codificar los datos de vídeo. Los bloques de vídeo pueden presentar tamaños fijos o variables y pueden diferir en tamaño de acuerdo con una norma de codificación especificada. Cada trama de vídeo incluye una serie de uno o más fragmentos. En la norma UIT-T H.264, por ejemplo, cada fragmento incluye una serie de macrobloques, que pueden estar dispuestos en subbloques. La norma H.264 admite la predicción intra en varios tamaños de bloque para codificación de vídeo bidimensional (2D), tales como 16 por 16, 8 por 8 o 4 por 4 para componentes de luma, y 8 por 8 para componentes de croma, así como la predicción inter en varios tamaños de bloque, tales como 16 por 16, 16 por 8, 8 por 16, 8 por 8, 8 por 4, 4 por 8 y 4 por 4 para componentes de luma y tamaños escalados correspondientes para componentes de croma. Los bloques de vídeo pueden comprender bloques de datos de píxel, o bloques de coeficientes de transformada, por ejemplo, tras un proceso de transformada tal como de transformada de coseno discreta (DCT) o un proceso de transformada conceptualmente similar. Estas técnicas pueden ampliarse al vídeo 3D.
- Los macrobloques 2D de la norma UIT-T H.264 pueden ampliarse a 3D codificando información de profundidad de un mapa de profundidad o mapa de paralaje junto con los componentes luma y croma asociados (es decir, componentes de textura) para esa trama o fragmento de vídeo. El mapeado por paralaje (también conocido como mapeado por desplazamiento virtual o mapeado offset) desplaza componentes de vista de textura situados en una ubicación de píxel basándose en una función de un ángulo de vista y un mapa de altura de la ubicación de píxel. El codificador de vídeo 22 puede codificar la información de profundidad como vídeo monocromático.
- Para codificar los bloques de vídeo, tales como un bloque codificado, el codificador de vídeo 22 realiza predicción intra o inter para generar uno o más bloques de predicción. El codificador de vídeo 22 resta los bloques de predicción de los bloques de vídeo originales que se van a codificar para generar bloques residuales. De este modo, los bloques residuales pueden representar diferencias de píxel a píxel entre los bloques que se codifican y los bloques de predicción. El codificador de vídeo 22 puede aplicar una transformada a los bloques residuales para generar bloques de coeficientes de transformada. Siguiendo las técnicas de codificación de predicción intra o inter y de transformada, el codificador de vídeo 22 puede cuantificar los coeficientes de transformada. La cuantificación se refiere en general a un proceso en el que los coeficientes se cuantifican para reducir posiblemente la cantidad de datos utilizados para representar los coeficientes. Después de la cuantificación, la codificación de entropía se puede

realizar de acuerdo con una metodología de codificación de entropía, tal como la codificación de longitud variable adaptativa al contexto (CAVLC) o la codificación aritmética binaria adaptativa al contexto (CABAC). A continuación se describen detalles adicionales de un proceso de codificación realizado por el codificador de vídeo 22 con respecto a la FIG. 2.

5 En la actualidad se están dedicando esfuerzos en la elaboración de una nueva norma de codificación de vídeo, conocida actualmente como codificación de vídeo de alta eficiencia (HEVC). La próxima norma también se conoce como H.265. Los trabajos de normalización se basan en un modelo de dispositivo de codificación de vídeo denominado modelo de prueba de HEVC (HM). El HM supone varias capacidades de los dispositivos de codificación
10 de vídeo respecto a dispositivos de acuerdo con, por ejemplo, la norma UIT-T H.264/AVC. Por ejemplo, mientras que la norma H.264 incluye nueve modalidades de codificación de predicción intra, el HM incluye hasta treinta y tres modalidades de codificación de predicción intra. La HEVC puede ampliarse para admitir las técnicas de información de cabecera de fragmento descritas en el presente documento.

15 El HM hace referencia a un bloque de datos de vídeo con el término unidad de codificación (CU). Los datos sintácticos de un flujo de bits pueden definir la unidad de codificación más grande (LCU), que es la unidad de codificación de mayor tamaño en lo que respecta al número de píxeles. En general, una CU tiene un propósito similar a un macrobloque de la norma H.264, excepto en que una CU no tiene una distinción de tamaño. Un bloque codificado puede ser una CU de acuerdo con la norma HM. Por lo tanto, una CU puede dividirse en varias sub-CU.
20 En general, las referencias de la presente divulgación a una CU pueden referirse a la unidad de codificación más grande (LCU) de una imagen o a una sub-CU de una LCU. Una LCU puede dividirse en varias sub-CU, y cada sub-CU puede dividirse en varias sub-CU. Los datos sintácticos para un flujo de bits pueden definir un número máximo de veces en que puede dividirse una LCU, que se denomina profundidad de CU. Por consiguiente, un flujo de bits también puede definir una unidad que es la unidad de codificación más pequeña (SCU). La presente divulgación
25 también utiliza el término «bloque» para referirse a cualquiera de entre una CU, una unidad de predicción (PU) o una unidad de transformada (TU).

Una LCU puede asociarse a una estructura de datos en árbol cuaternario. En general, una estructura de datos en árbol cuaternario incluye un nodo por cada CU, donde un nodo raíz corresponde a la LCU. Si una CU se divide en
30 cuatro sub-CU, el nodo correspondiente a la CU incluye cuatro nodos hoja, cada uno de los cuales corresponde a una de las sub-CU. Cada nodo de la estructura de datos en árbol cuaternario puede incluir datos sintácticos para la CU correspondiente. Por ejemplo, un nodo del árbol cuaternario puede incluir un indicador de división, que indica si la CU correspondiente al nodo está dividida o no en varias sub-CU. Los elementos sintácticos para una CU pueden definirse de manera recursiva y pueden depender de si la CU está dividida o no en varias sub-CU.

35 Una CU que no está dividida puede incluir una o más unidades de predicción (PU). En general, una PU representa la totalidad o una parte de la CU correspondiente, e incluye datos para recuperar una muestra de referencia para la PU. Por ejemplo, cuando la PU está codificada en modalidad intra, la PU puede incluir datos que describen una modalidad de predicción intra para la PU. En otro ejemplo, cuando la PU está codificada en la modalidad inter, la PU
40 puede incluir datos que definen un vector de movimiento para la PU. Los datos que definen el vector de movimiento pueden describir, por ejemplo, un componente horizontal del vector de movimiento, un componente vertical del vector de movimiento, una resolución para el vector de movimiento (por ejemplo, con una precisión de un cuarto de píxel o una precisión de un octavo de píxel), una trama de referencia a la que apunta el vector de movimiento y/o una lista de referencia (por ejemplo, lista 0 o lista 1) para el vector de movimiento. El vector de movimiento también
45 puede tratarse como si tuviera resoluciones diferentes para los componentes de vista de textura y los componentes de vista de profundidad. Los datos para la CU que definen la(s) PU también pueden describir, por ejemplo, una división de la CU en una o más PU. Las modalidades de división pueden diferir en función de si la CU está sin codificar, codificada en la modalidad de predicción intra o codificada en la modalidad de predicción inter.

50 Una CU que presenta una o más PU también puede incluir una o más unidades de transformada (TU). Tras la predicción mediante una PU, el codificador de vídeo 22 puede calcular un valor residual para la parte de la CU correspondiente a la PU. El valor residual puede transformarse, examinarse y cuantificarse. Una TU no está necesariamente limitada al tamaño de una PU. Por lo tanto, las TU pueden ser más grandes o más pequeñas que las PU correspondientes para la misma CU. En algunos ejemplos, el tamaño máximo de una TU puede corresponder
55 al tamaño de la CU correspondiente.

Como se ha señalado anteriormente, la predicción intra incluye la predicción de una PU de una CU actual de una imagen procedente de unas CU codificadas previamente de la misma imagen. Más específicamente, el codificador de vídeo 22 puede realizar la predicción intra de una CU actual de una imagen, mediante una modalidad de
60 predicción intra particular. Un codificador HM puede configurarse con hasta treinta y tres modalidades de predicción intra. Por lo tanto, para admitir un mapeado uno a uno entre las modalidades de predicción intra direccional y las transformadas direccionales, los codificadores y decodificadores HM necesitarían almacenar 66 matrices para cada tamaño de transformada admitido. Además, los tamaños de bloque para los cuales se admiten las treinta y tres modalidades de predicción intra pueden ser bloques relativamente grandes, por ejemplo, de 32x32 píxeles, 64x64
65 píxeles o incluso de mayor tamaño.

En el dispositivo de destino 16, el decodificador de vídeo 28 recibe datos de vídeo codificados 8. El decodificador de vídeo 28 realiza la decodificación de entropía de los datos de vídeo codificados recibidos 8, tales como un bloque codificado, de acuerdo con una metodología de codificación de entropía, tal como la CAVLC o la CABAC, para obtener los coeficientes cuantificados. El decodificador de vídeo 28 aplica las funciones de cuantificación inversa (decuantificación) y de transformada inversa para reconstruir el bloque residual en el dominio del píxel. El decodificador 28 de vídeo genera también un bloque de predicción basado en información de control o información de sintaxis (por ejemplo, modalidad de codificación, vectores de movimiento, sintaxis que define coeficientes de filtro y similares) incluidos en los datos de vídeo codificados. El decodificador de vídeo 28 calcula una suma del bloque de predicción y el bloque residual reconstruido para generar un bloque de vídeo reconstruido para su visualización. A continuación se describen detalles adicionales de un ejemplo de proceso de decodificación realizado por el decodificador de vídeo 28, con respecto a la FIG. 5.

Como se describe en el presente documento, Y puede representar luminancia, Cb y Cr pueden representar dos valores diferentes de crominancia de un espacio de color YCbCr tridimensional (por ejemplo, tonalidades azules y rojas), y D puede representar información de profundidad. En algunos ejemplos, cada ubicación de píxel puede definir realmente tres valores de píxel para un espacio de color tridimensional y un valor de píxel para la profundidad de la ubicación de píxel. En otros ejemplos, puede haber números diferentes de componentes de luma por cada componente de croma. Por ejemplo, puede haber cuatro componentes de luma por cada componente de croma. Además, los componentes de profundidad y textura pueden tener resoluciones diferentes. En dicho ejemplo, puede no haber una relación de uno a uno entre los componentes de vista de textura (por ejemplo, componentes de luma) y los componentes de vista de profundidad. Sin embargo, las técnicas de la presente divulgación pueden referirse a la predicción con respecto a una dimensión a fin de simplificar. En la medida en que se describen técnicas con respecto a los valores de píxel en una dimensión, se pueden ampliar técnicas similares a las otras dimensiones. En particular, de acuerdo con un aspecto de la presente divulgación, el codificador de vídeo 22 y/o el decodificador de vídeo 28 pueden obtener un bloque de píxeles, en el que el bloque de píxeles incluye componentes de vista de textura y componentes de vista de profundidad.

En algunos ejemplos, el codificador de vídeo 22 y el decodificador de vídeo 28 pueden utilizar una o más técnicas de filtrado de interpolación durante la compensación de movimiento. Es decir, el codificador de vídeo 22 y/o el decodificador de vídeo 28 pueden aplicar un filtro de interpolación al soporte de filtro que comprende conjuntos de posiciones de píxel entero.

El decodificador de vídeo 28 del dispositivo de destino 16 recibe uno o más bloques codificados como parte de un flujo de bits de vídeo codificado junto con información adicional, incluidos los elementos sintácticos relacionados con los componentes de vista de textura. El decodificador 28 de vídeo puede renderizar datos de vídeo para la reproducción 3D, basándose en el bloque codificado 8 y los elementos sintácticos. De acuerdo con las técnicas de la presente divulgación, y como se analiza con mayor detalle a continuación, se pueden utilizar elementos sintácticos señalados para componentes de vista de textura 4 a fin de predecir elementos sintácticos para componentes de vista de profundidad 6. Los elementos sintácticos se pueden señalar en una cabecera de fragmento para componentes de vista de textura 4. Los elementos sintácticos correspondientes para los componentes de vista de profundidad 6 pueden determinarse a partir de los elementos sintácticos relacionados para los componentes de vista de textura 4.

Algunos elementos sintácticos para los componentes de vista de profundidad 6 pueden señalarse en una cabecera de fragmento para componentes de vista de textura 4. De forma alternativa, se pueden señalar algunos elementos sintácticos para los componentes de vista de textura 4 en una cabecera de fragmento para componentes de vista de profundidad 6. En un ejemplo, el codificador de vídeo 22 del dispositivo de origen 12 puede configurarse para predecir un elemento sintáctico para una cabecera de fragmento de una vista actual a partir de una cabecera de fragmento de referencia, y codificar la vista actual basándose en los elementos sintácticos predichos. En otro ejemplo de la divulgación, el decodificador de vídeo 30 del dispositivo de destino 16 puede configurarse para predecir un elemento sintáctico para una cabecera de fragmento de una vista actual a partir de una cabecera de fragmento de referencia, y codificar la vista actual basándose en los elementos sintácticos predichos.

Las técnicas de la presente divulgación describen una extensión 3DV de cabecera de fragmento. Las técnicas pueden aplicarse para predecir elementos sintácticos para un componente de vista de profundidad, a partir de elementos sintácticos almacenados en una cabecera de fragmento para componentes de vista de textura coubicados de la misma vista y de la misma unidad de acceso. La unidad de acceso puede ser una unidad de capa de abstracción de red (NAL) que incluye componentes de vista de textura y componentes de vista de profundidad correspondientes. La predicción de los elementos sintácticos puede controlarse en el nivel de cabecera de fragmento. Algunos elementos sintácticos de una cabecera de fragmento del componente de vista de profundidad o del componente de vista de textura pueden predecirse a partir de una cabecera de fragmento de referencia.

En un ejemplo, la cabecera de fragmento de referencia es del componente de vista de la misma vista de la misma unidad de acceso que los elementos sintácticos predichos o es de la textura de vista básica compatible con la codificación de vídeo avanzada (AVC) de la misma unidad de acceso. De forma alternativa, la cabecera de fragmento de referencia de un componente de vista de profundidad es la cabecera de fragmento del componente de

vista de textura de la misma vista o la cabecera de fragmento del componente de vista de profundidad de la primera vista por orden de decodificación. Se puede señalar un ID de índice de orden de vista delta para indicar a partir de qué vista se predice la cabecera de fragmento, en la misma unidad de acceso. En otros ejemplos, se puede señalar un indicador que indica si la cabecera de fragmento de referencia es del componente de vista de profundidad o del
5 componente de vista de textura. La vista específica también se puede señalar en un flujo de bits codificado.

Otras extensiones de la predicción de cabecera de fragmento pueden incluir organizar los elementos sintácticos de una cabecera de fragmento en un orden diferente al utilizado para los elementos sintácticos de cabecera de fragmento en una cabecera de fragmento de AVC. En otros ejemplos, un indicador de control de cada extensión
10 3DV de cabecera de fragmento puede indicar si se han predicho slice_type, num_ref_idx_I0_active_minus1, num_ref_idx_I1_active_minus1 y la modificación de la lista de imágenes de referencia. En algunas técnicas, cuando se habilita la predicción de la modificación de la lista de imágenes de referencia, la modificación de lista de imágenes de referencia puede refinarse basándose en una tabla de sintaxis de exclusión o inserción de lista de imágenes de referencia recién introducida.

En algunos ejemplos, los indicadores u otras indicaciones descritos anteriormente se señalan solamente en la extensión 3DV de la cabecera de fragmento. Después de que se indique un componente de vista de profundidad o textura, se puede señalar además un ID de fragmento del fragmento de un componente de vista dado en una unidad de acceso. Además, se puede señalar el ID de fragmento para cada fragmento de cada componente de vista, ya sea
20 en la cabecera de fragmento o en una cabecera de unidad de capa de abstracción de red (NAL) correspondiente.

La FIG. 2 es un diagrama de bloques que ilustra con mayor detalle un ejemplo del codificador de vídeo 22 de la FIG. 1. El codificador de vídeo 22 codifica unidades de bloque que señalan elementos sintácticos para componentes de vista de textura que pueden utilizarse para predecir elementos sintácticos para componentes de vista de profundidad, de conformidad con las técnicas de la presente divulgación. El codificador de vídeo 22 es un ejemplo de dispositivo o aparato informático de vídeo, denominado «codificador» en el presente documento. Como se muestra en la FIG. 2, el codificador de vídeo 22 corresponde al codificador de vídeo 22 del dispositivo de origen 12. Sin embargo, en otros ejemplos, el codificador de vídeo 22 puede corresponder a un dispositivo diferente. En otros ejemplos, otras unidades (tales como, por ejemplo, otro codificador/decodificador (CÓDEC)) también pueden realizar técnicas similares a las realizadas por el codificador de vídeo 22.
25 30

El codificador de vídeo 22 puede realizar al menos una de entre la codificación intra y la codificación inter de bloques dentro de tramas de vídeo, aunque los componentes de codificación intra no se muestran en la FIG. 2 a fin de facilitar la ilustración. La codificación intra se basa en la predicción espacial para reducir o eliminar la redundancia espacial en el vídeo de una trama de vídeo dada. La codificación inter se basa en la predicción temporal para reducir o eliminar la redundancia temporal en el vídeo de tramas adyacentes de una secuencia de vídeo. La modalidad intra (modalidad I) puede referirse a la modalidad de compresión espacial. Las modalidades inter, tales como una predicción unidireccional (modalidad P) o una predicción bidireccional (modalidad B), pueden referirse a las modalidades de compresión temporal. Las técnicas de la presente divulgación se aplican durante la codificación inter y la codificación intra. Sin embargo, para simplificar y facilitar la ilustración, las unidades intracodificadas, tales como una unidad de predicción espacial, no se ilustran en la FIG. 2.
35 40

Como se muestra en la FIG. 2, el codificador de vídeo 22 recibe un bloque de vídeo dentro de una trama de vídeo que se va a codificar. En un ejemplo, el codificador de vídeo 22 recibe componentes de vista de textura 4 y componentes de vista de profundidad 6. En otro ejemplo, el codificador de vídeo recibe la vista 2 desde la fuente de vídeo 20.
45

En el ejemplo de la FIG. 2, el codificador de vídeo 22 incluye una unidad de codificación de predicción (MCU) en una unidad de procesamiento de predicción 32, una unidad de vídeo multivista más profundidad (MVD) 33, una memoria 34, un primer sumador 48, una unidad de procesamiento de transformada 38, una unidad de cuantificación 40 y una unidad de codificación de entropía 46. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 22 incluye además una unidad de cuantificación inversa 42, una unidad de procesamiento de transformada inversa 44, un segundo sumador 51 y una unidad de eliminación de bloques 43. La unidad de eliminación de bloques 43 es un filtro de eliminación de bloques para filtrar los límites de bloque a fin de eliminar distorsiones de efecto pixelado del vídeo reconstruido. Si está incluida en el codificador de vídeo 22, la unidad de eliminación de bloques 43 típicamente filtrará la salida del segundo sumador 51. La unidad de eliminación de bloques 43 puede determinar información de eliminación de bloques para el uno o más componentes de vista de textura. La unidad de eliminación de bloques 43 también puede determinar información de eliminación de bloques para el componente de mapa de profundidad. En algunos ejemplos, la información de eliminación de bloques para el uno o más componentes de textura puede ser diferente de la información de eliminación de bloques para el componente de mapa de profundidad. En un ejemplo, mostrado en la figura 2, la unidad de procesamiento de transformada 38 representa un bloque funcional, al contrario que una «TU» con respecto a la HEVC.
50 55 60

La unidad de vídeo multivista más profundidad (MVD) 33 recibe uno o más bloques de vídeo (designados por «BLOQUE DE VÍDEO» en la FIG. 2) que comprenden componentes de textura e información de profundidad, tales como los componentes de vista de textura 4 y componentes de vista de profundidad 6. La unidad de MVD 33
65

proporciona funcionalidad al codificador de vídeo 22 para codificar componentes de profundidad en una unidad de bloque. La unidad de MVD 33 envía los componentes de vista de textura y los componentes de vista de profundidad, combinados o por separado, a la unidad de procesamiento de predicción 32 en un formato que permite a la unidad de procesamiento de predicción 32 procesar información de profundidad. La unidad de MVD 33 también puede

5 señalar a la unidad de procesamiento de transformada 38 que los componentes de vista de profundidad se incluyen con el bloque de vídeo. En otros ejemplos, cada unidad del codificador de vídeo 22, tal como la unidad de procesamiento de predicción 32, la unidad de procesamiento de transformada 38, la unidad de cuantificación 40, la unidad de codificación de entropía 46, etc., comprende funciones para procesar información de profundidad además de componentes de vista de textura.

10 En general, el codificador de vídeo 22 codifica la información de profundidad de una manera similar a la información de crominancia, en la medida en que la unidad de compensación de movimiento 37 está configurada para reutilizar vectores de movimiento calculados para un componente de luminancia de un bloque, al calcular un valor predicho para un componente de profundidad del mismo bloque. De forma similar, una unidad de predicción intra del

15 codificador de vídeo 22 puede estar configurada para utilizar una modalidad de predicción intra seleccionada para el componente de luminancia (es decir, basada en el análisis del componente de luminancia), al codificar el componente de vista de profundidad mediante predicción intra.

20 La unidad de procesamiento de predicción 32 incluye una unidad de estimación de movimiento (ME) 35 y una unidad de compensación de movimiento (MC) 37. La unidad de procesamiento de predicción 32 predice información de profundidad para ubicaciones de píxel, así como para componentes de textura. Uno o más filtros de interpolación 39 (denominados «filtro 39» en el presente documento) pueden estar incluidos en la unidad de procesamiento de predicción 32, y una o ambas de las unidades de ME 35 y MC 37 pueden recurrir a estos para realizar la interpolación como parte de la estimación de movimiento y/o la compensación de movimiento. El filtro de

25 interpolación 39 puede representar en realidad una pluralidad de filtros diferentes para facilitar numerosos tipos diferentes de interpolación y filtrado tipo interpolación. Por lo tanto, la unidad de procesamiento de predicción 32 puede incluir una pluralidad de filtros de interpolación o de filtros tipo interpolación.

30 Durante el proceso de codificación, el codificador de vídeo 22 recibe un bloque de vídeo para codificar (designado por «BLOQUE DE VÍDEO» en la FIG. 2), y la unidad de procesamiento de predicción 32 realiza la codificación de predicción inter para generar un bloque de predicción (designado por «BLOQUE DE PREDICCIÓN» en la FIG. 2). El bloque de predicción incluye tanto componentes de vista de textura como información de vista de profundidad. Específicamente, la unidad de ME 35 puede realizar una estimación de movimiento para identificar el bloque de predicción en la memoria 34, y la unidad de MC 37 puede realizar una compensación de movimiento para generar el

35 bloque de predicción.

Se considera típicamente que la estimación de movimiento es el proceso de generar vectores de movimiento, que estiman un movimiento para bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de un bloque de predicción de una trama de predicción o referencia (u otra unidad codificada, por

40 ejemplo, un fragmento) con respecto al bloque que se va a codificar en la trama actual (u otra unidad codificada). El vector de movimiento puede tener una precisión de píxel entero o subentero. Por ejemplo, tanto un componente horizontal como un componente vertical del vector de movimiento pueden tener componentes enteros y componentes subenteros respectivos. La trama de referencia (o parte de la trama) puede estar situada temporalmente antes o después de la trama de vídeo (o parte de la trama de vídeo) a la que pertenece el bloque de

45 vídeo actual. La compensación de movimiento se considera típicamente el proceso de extraer o generar el bloque de predicción de la memoria 34, que puede incluir interpolar o generar de otro modo los datos predictivos, basándose en el vector de movimiento determinado mediante la estimación de movimiento.

50 La unidad de ME 35 calcula al menos un vector de movimiento para el bloque de vídeo que se va a codificar, comparando el bloque de vídeo con los bloques de referencia de una o más tramas de referencia (por ejemplo, una trama previa y/o posterior). Los datos para las tramas de referencia pueden almacenarse en la memoria 34. La unidad de ME 35 puede realizar la estimación de movimiento con precisión de píxel fraccionario, a veces denominada estimación de movimiento de píxel fraccionario, por fraccionario, subentero o subpíxel. En la estimación de movimiento de píxel fraccionario, la unidad de ME 35 calcula un vector de movimiento que indica un desplazamiento a una ubicación distinta de una ubicación de píxel entero. Por lo tanto, el vector de movimiento puede tener una precisión de píxel fraccionario, por ejemplo, precisión de medio píxel, precisión de un cuarto de píxel, precisión de un octavo de píxel u otras precisiones de píxel fraccionario. De esta manera, la estimación de movimiento de píxel fraccionario permite a la unidad de procesamiento de predicción 32 estimar el movimiento con mayor precisión que las ubicaciones de píxel entero (o píxel completo) y, por tanto, la unidad de procesamiento de

55 predicción 32 genera un bloque de predicción más preciso. La estimación de movimiento de píxel fraccionario permite a la unidad de procesamiento de predicción 32 predecir información de profundidad con una primera resolución y predecir los componentes de textura con una segunda resolución. Por ejemplo, los componentes de textura se predicen con una precisión de píxel completo, mientras que la información de profundidad se predice con una precisión de medio píxel. En otros ejemplos, pueden utilizarse otras resoluciones del vector de movimiento para

60 la información de profundidad y los componentes de textura.

65

La unidad de ME 35 puede recurrir a uno o más filtros 39 para cualquier interpolación necesaria durante el proceso de estimación de movimiento. En algunos ejemplos, la memoria 34 puede almacenar valores interpolados para píxeles subenteros, que pueden ser calculados, por ejemplo, por el sumador 51 mediante filtros 39. Por ejemplo, el sumador 51 puede aplicar filtros 39 a bloques reconstruidos que van a almacenarse en la memoria 34.

5 Una vez que la unidad de procesamiento de predicción 32 ha generado el bloque de predicción, el codificador de vídeo 22 forma un bloque de vídeo residual (designado por «BLOQUE RESID.» en la FIG. 2), restando el bloque de predicción del bloque de vídeo original que se está codificando. Esta sustracción puede ocurrir entre componentes de textura del bloque de vídeo original y componentes de textura del bloque de predicción, así como para la información de profundidad del bloque de vídeo original o el mapa de profundidad de la información de profundidad situada en el bloque de predicción. El sumador 48 representa el componente o los componentes que realizan esta operación de sustracción.

15 La unidad de procesamiento de transformada 38 aplica una transformada, tal como una transformada discreta del coseno (DCT) o una transformada conceptualmente similar, al bloque residual, generando un bloque de vídeo que comprende coeficientes de bloque de transformada residual. Debe entenderse que la unidad de procesamiento de transformada 38 representa el componente del codificador de vídeo 22 que aplica una transformada a los coeficientes residuales de un bloque de datos de vídeo, al contrario que una TU de una CU definida por la HEVC. La unidad de procesamiento de transformada 38 puede llevar a cabo otras transformadas, tales como las definidas por la norma H.264, que son conceptualmente similares a la DCT. Dichas transformadas incluyen, por ejemplo, transformadas direccionales (tales como transformadas de teorema de Karhunen-Loeve), transformadas de ondícula, transformadas enteras, transformadas de subbanda u otros tipos de transformadas. En cualquier caso, la unidad de procesamiento de transformada 38 aplica la transformada al bloque residual, generando un bloque de coeficientes de transformada residuales. La unidad de procesamiento de transformada 38 puede aplicar el mismo tipo de transformada tanto a los componentes de textura como a la información de profundidad de bloques residuales correspondientes. Habrá bloques residuales separados para cada componente de textura y profundidad. La transformada convierte la información residual de un dominio de píxel a un dominio de frecuencia.

30 La unidad de cuantificación 40 cuantifica los coeficientes de transformada residuales para reducir todavía más la velocidad de bits. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos o todos los coeficientes. La unidad de cuantificación 40 puede cuantificar un residuo de codificación de imagen de profundidad. Tras la cuantificación, la unidad de codificación de entropía 46 realiza la codificación de entropía de los coeficientes de transformada cuantificados. Por ejemplo, la unidad de codificación de entropía 46 puede realizar la CAVLC, la CABAC u otro procedimiento de codificación de entropía.

35 La unidad de codificación de entropía 46 puede codificar también uno o más vectores de movimiento e información de soporte obtenida de la unidad de procesamiento de predicción 32 u otro componente del codificador de vídeo 22, tal como la unidad de cuantificación 40. El uno o más elementos sintácticos de predicción pueden incluir una modalidad de codificación, datos para uno o más vectores de movimiento (por ejemplo, componentes horizontales y verticales, identificadores de lista de referencia, índices de lista y/o información de señalización de resolución de vector de movimiento), una indicación de una técnica de interpolación utilizada, un conjunto de coeficientes de filtro, una indicación de la resolución relativa de la imagen de profundidad para la resolución del componente de luma, una matriz de cuantificación para el residuo de codificación de imagen de profundidad, información de eliminación de bloques para la imagen de profundidad u otra información asociada con la generación del bloque de predicción. Estos elementos sintácticos de predicción pueden incluirse en el nivel de secuencia o en el nivel de imagen.

50 El uno o más elementos sintácticos pueden incluir también una diferencia de parámetro de cuantificación (QP) entre el componente de luma y el componente de profundidad. La diferencia QP se puede señalar en el nivel de fragmento y se puede incluir en una cabecera de fragmento para los componentes de vista de textura. También pueden señalarse otros elementos sintácticos en un nivel de unidad de bloque de bloque codificado, que incluye un patrón de bloque codificado para el componente de vista de profundidad, un QP delta para el componente de vista de profundidad, una diferencia de vector de movimiento u otra información asociada con la generación del bloque de predicción. La diferencia de vector de movimiento puede señalarse como un valor delta entre un vector de movimiento objetivo y un vector de movimiento de los componentes de textura, o como un valor delta entre el vector de movimiento objetivo (es decir, el vector de movimiento del bloque que se codifica) y un predictor de vectores de movimiento contiguos para el bloque (por ejemplo, una PU de una CU). Tras la codificación de entropía realizada por la unidad de codificación de entropía 46, el vídeo codificado y los elementos sintácticos pueden transmitirse a otro dispositivo o archivarse (por ejemplo, en la memoria 34) para su posterior transmisión o recuperación.

60 La unidad de cuantificación inversa 42 y la unidad de procesamiento de transformada inversa 44 aplican la cuantificación inversa y la transformada inversa, respectivamente, para reconstruir el bloque residual en el dominio del píxel, por ejemplo, para su posterior uso como bloque de referencia. El bloque residual reconstruido (designado por «BLOQUE RESID. RECON.» en la FIG. 2) puede representar una versión reconstruida del bloque residual enviado a la unidad de procesamiento de transformada 38. El bloque residual reconstruido puede diferir del bloque residual generado por el sumador 48, debido a la pérdida de detalle causada por las operaciones de cuantificación y cuantificación inversa. El sumador 51 suma el bloque residual reconstruido al bloque de predicción con

compensación de movimiento, generado por la unidad de procesamiento de predicción 32, para generar un bloque de vídeo reconstruido para su almacenamiento en la memoria 34. La unidad de procesamiento de predicción 32 puede utilizar el bloque de vídeo reconstruido como bloque de referencia que se puede utilizar para codificar posteriormente una unidad de bloque en una trama de vídeo subsiguiente o una unidad codificada subsiguiente.

5 De esta manera, el codificador de vídeo 22 representa un ejemplo de codificador de vídeo configurado para recibir una unidad de bloque codificado que comprende un componente de vista indicativo de una vista de una imagen, en el que el componente de vista comprende uno o más componentes de vista de textura y un componente de vista de profundidad, generar una cabecera de fragmento de textura para el uno o más componentes de vista de textura que incluyen elementos sintácticos de textura, en el que se pueden determinar elementos sintácticos de profundidad para el componente de vista de profundidad a partir de los elementos sintácticos de textura de la cabecera de fragmento de textura.

15 En algunos casos, la información relativa a la codificación de los componentes de vista de textura y los componentes de vista de profundidad se indican como uno o más elementos sintácticos para su inclusión en el flujo de bits codificado. En algunos ejemplos, una cabecera de fragmento de profundidad comprende elementos sintácticos que incluyen al menos uno de entre la ubicación de dirección de fragmento del macrobloque inicial, el tipo de fragmento, el conjunto de parámetros de imagen (PPS) que se va a utilizar, el QP delta entre el QP inicial del fragmento y el QP señalado en el PPS, el orden de las imágenes de referencia (representadas como frame_num) y un orden de visualización de la imagen actual (POC). La cabecera de fragmento de profundidad también puede comprender al menos uno de entre una construcción de lista de imágenes de referencia y elementos sintácticos relacionados, una operación de control de gestión de memoria y elementos sintácticos relacionados, y una predicción ponderada y elementos sintácticos relacionados.

25 La FIG. 3 es un diagrama conceptual que ilustra un ejemplo de orden de decodificación multivista, de acuerdo con las técnicas de la presente divulgación; Se muestra un orden de decodificación MVC típico (es decir, un orden de flujo de bits) en la FIG. 3. La disposición de orden de decodificación se denomina codificación centrada en el tiempo. Cada unidad de acceso se define para contener las imágenes codificadas de todas las vistas para un instante de tiempo de salida. Debe tenerse en cuenta que el orden de las unidades de acceso puede no ser idéntico al orden de salida o de visualización.

35 La FIG. 4 es un diagrama de un ejemplo de estructura de predicción MVC (MVC) para codificación de vídeo multivista. La MVC es una extensión de la norma H.264/AVC. La FIG. 4 ilustra una estructura de predicción MVC típica (que incluye tanto la predicción inter de imágenes dentro de cada vista como la predicción inter de vistas) para la codificación de vídeo multivista. La estructura de predicción MVC incluye tanto la predicción inter de imágenes dentro de cada vista como la predicción inter de vistas. En la FIG. 4, las predicciones se indican mediante flechas, donde el objeto al que se apunta utiliza el objeto desde el que se apunta como referencia de predicción. La estructura de predicción MVC de la FIG. 4 puede utilizarse junto con una disposición de orden de decodificación centrada en el tiempo. En un orden de decodificación centrada en el tiempo, cada unidad de acceso puede definirse para contener imágenes codificadas de todas las vistas para un instante de tiempo de salida. El orden de decodificación de las unidades de acceso puede no ser idéntico al orden de salida o de visualización.

45 En la MVC, la predicción inter de vistas se admite gracias a la compensación de movimiento de disparidad, que utiliza la sintaxis de la compensación de movimiento H.264/AVC, pero permite que una imagen de una vista diferente se disponga como imagen de referencia. La MVC podría admitir también la codificación de dos vistas. Un codificador MVC puede obtener más de dos vistas como entrada de vídeo 3D, y un decodificador MVC puede decodificar una representación multivista. Un renderizador con un decodificador MVC puede decodificar contenido de vídeo 3D con varias vistas.

50 Las imágenes de la misma unidad de acceso (es decir, con el mismo instante de tiempo) se pueden someter a predicción inter de vistas en MVC. Cuando se codifica una imagen de una de las vistas no básicas, se puede añadir una imagen a una lista de imágenes de referencia si está en una vista diferente pero presenta un mismo instante de tiempo. Una imagen de referencia de predicción inter de vistas puede disponerse en cualquier posición de una lista de imágenes de referencia, tal como cualquier imagen de referencia de predicción inter.

55 Como se muestra en la FIG. 4, un componente de vista puede utilizar los componentes de vista de otras vistas como referencia. En MVC, la predicción inter de vistas puede realizarse como si el componente de vista de otra vista fuera una referencia de predicción inter. Las potenciales referencias inter de vistas pueden señalarse en la extensión MVC del conjunto de parámetros de secuencia (SPS). Las referencias potenciales inter de vistas pueden modificarse mediante el proceso de construcción de la lista de imágenes de referencia, lo cual permite el ordenamiento flexible de las referencias de predicción inter o de predicción inter de vistas. La tabla 1 siguiente muestra un ejemplo de extensión MVC de SPS.

Tabla 1 – Ejemplo de extensión MVC de SPS

seq_parameter_set_mvc_extension() {	C	Descriptor
-------------------------------------	----------	-------------------

num_views_minus1	0	ue(v)
for(i = 0; i <= num_views_minus1; i++)		
view_id[i]	0	ue(v)
for(i = 1; i <= num_views_minus1; {		
num_anchor_refs_l0 [i]	0	ue(v)
for(j = 0; j < num_anchor_refs_l0[i]; j++)		
anchor_ref_l0[i][j]	0	ue(v)
num_anchor_refs_l1 [i]	0	ue(v)
for(j = 0; j < num_anchor_refs_l1[i]; j++)		
anchor_ref_l1[i][j]	0	ue(v)
}		
for(i = 1; i <= num_views_minus1; i++ {		
num_non_anchor_refs_l0[i]	0	ue(v)
for(j = 0; j < num_non_anchor_refs_l0[i]; j++)		
non_anchor_ref_l0[i][j]	0	ue(v)
num_non_anchor_refs_l1 [i]	0	ue(v)
for(j = 0; j < num_non_anchor_refs_l1[i]; j++)		
non_anchor_ref_l1	0	ue(v)
}		
num_level_values_signalled_minus1	0	ue(v)
for(i = 0;		
i <= num level values signalled minus1; {		
level_idc[i]	0	u(8)
num_applicable_ops_minus1[i]	0	ue(v)
for(j = 0; j <= num_applicable_ops_minus1[i]; j++ {		
applicable_op_temporal_id[i][j]	0	u(3)
applicable_op_num_target_views_minus1[i][j]	0	ue(v)
for(k = 0; k <= applicable_op_num_target_views_minus1[i][j]; k++)		
applicable_op_target_view_id[i][j][k]	0	ue(v)
applicable_op_num_views_minus1[i][j]	0	ue(v)
}		
}		
}		

5 Por el contrario, en la HEVC, la cabecera de fragmento sigue un principio de diseño similar a la H.264/AVC. Conceptualmente, tanto la HEVC como la H.264/AVC incluyen una capa de codificación de vídeo (VCL) y una capa de abstracción de red (NAL). La VCL incluye todo el procesamiento de señal de bajo nivel, incluida la división de bloques, la predicción inter e intra, la codificación basada en transformada, la codificación de entropía, el filtrado en bucle y similares. La NAL encapsula datos codificados e información asociada en unidades NAL, formato que es apropiado para los sistemas de transmisión de vídeo y de aplicaciones. Además, una cabecera de fragmento HEVC puede contener una sintaxis de parámetros de filtro de bucle adaptativo (ALF) en la especificación actual de la HEVC. En algunos ejemplos, la cabecera de fragmento de profundidad comprende uno o más parámetros de filtro de bucle adaptativo.

10 De modo similar a la H.264/AVC, un flujo de bits HEVC incluye un número de unidades de acceso, incluyendo cada unidad de acceso datos codificados asociados con una imagen que tiene un tiempo de captación o presentación diferenciado.

15 Cada unidad de acceso está dividida en unidades NAL, que incluyen una o más unidades NAL VCL (es decir, unidades NAL de fragmento codificadas) y cero o más unidades NAL no VCL, por ejemplo, unidades NAL de conjunto de parámetros o unidades NAL de información de mejora complementaria (SEI). Cada unidad NAL incluye

una cabecera de unidad NAL y una carga útil de unidad NAL. A la información de la cabecera de la unidad NAL puede accederse mediante pasarelas de medios, también conocidas como elementos de red sensibles a los medios (MANE), para operaciones inteligentes sensibles a los medios de transmisión, como la adaptación a la transmisión.

5 Una diferencia de la HEVC en comparación con la H.264/AVC es la estructura de codificación dentro de una imagen. En la HEVC, cada imagen se divide en bloques en árbol de hasta 64x64 muestras de luma. Los bloques en árbol pueden dividirse recursivamente en unidades de codificación (CU) más pequeñas mediante una estructura genérica de segmentación en árbol cuaternario. Las CU pueden dividirse además en unidades de predicción (PU) y unidades de transformada (TU). Las unidades de predicción (PU) pueden utilizarse para la predicción intra e inter. Las
10 unidades de transformada pueden definirse para transformar y cuantificar. La HEVC incluye transformadas enteras para un número de tamaños de TU.

Las imágenes codificadas mediante predicción pueden incluir fragmentos sometidos a predicción unidireccional y a predicción bidireccional. La flexibilidad en la creación de estructuras de codificación de imagen es comparable
15 aproximadamente a la de la H.264/AVC. La VCL en un codificador HEVC genera, y en un decodificador HEVC consume, estructuras sintácticas diseñadas para ser adaptables a los tamaños de unidad de transmisión máxima (MTU) que se encuentran comúnmente en las redes IP, independientemente del tamaño de una imagen codificada. La segmentación de imágenes se puede conseguir a través de fragmentos. Se incluye un concepto de «fragmentos de granularidad fina» (FGS) que permite la creación de límites de fragmento dentro de un bloque en árbol.

20 Como en la H.264/AVC, se puede aplicar un filtrado de eliminación de bloques en bucle a la imagen reconstruida. La HEVC también incluye dos filtros en bucle que pueden aplicarse después del filtrado de eliminación de bloques: Offset de muestra adaptativo (SAO) y filtro de bucle adaptativo (ALF).

25 Otra diferencia de la HEVC en comparación con la H.264/AVC es la disponibilidad de herramientas de codificación basadas en VCL que están diseñadas para permitir el procesamiento en arquitecturas paralelas de nivel alto. Se pueden utilizar fragmentos corrientes, como en la HEVC y la H.264/AVC, para propósitos de procesamiento paralelo. Además de los fragmentos corrientes, están disponibles tres nuevas herramientas de procesamiento en paralelo, a saber, los fragmentos de entropía, los elementos y el procesamiento paralelo de frente de onda.

30 La norma de codificación de vídeo de alta eficiencia (HEVC) actualmente en elaboración también utiliza cabeceras de fragmento. En la HEVC, la cabecera de fragmento sigue el mismo principio de diseño que en la H.264/AVC. La cabecera de fragmento de HEVC también contiene la sintaxis de parámetros de filtro de bucle adaptativo (ALF). Sin embargo, es discutible que los parámetros del filtro de bucle adaptativo deban disponerse en la cabecera de
35 fragmento.

La construcción actual de la cabecera de fragmento en la H.264/MVC y la HEVC presenta algunos inconvenientes. Por ejemplo, algunos elementos sintácticos no son ligeros, y permitir que las técnicas de predicción se utilicen para todos los elementos sintácticos «no ligeros» puede no ser beneficioso. Por ejemplo, la predicción de toda la tabla de
40 sintaxis de modificación de lista de imágenes de referencia puede no ser beneficiosa. Además, la predicción de cabecera de fragmento entre los mismos componentes de vista puede no siempre aportar la eficiencia deseable.

En el contexto de las técnicas de la presente divulgación, la predicción de la sintaxis para elementos sintácticos de cabecera de fragmento puede implicar copiar o insertar elementos sintácticos de otra cabecera de fragmento (por
45 ejemplo, una cabecera de fragmento de referencia) para su uso por una cabecera de fragmento actual. Los elementos sintácticos que se van a copiar o insertar pueden designarse mediante un indicador o un índice. Además, un indicador o índice puede designar una cabecera de fragmento de referencia.

En un códec 3DV, un componente de vista de cada vista de un instante de tiempo específico puede incluir un
50 componente de vista de textura y un componente de vista de profundidad. Se puede utilizar una estructura de fragmento para propósitos de recuperación frente a errores. Sin embargo, un componente de vista de profundidad solo puede ser significativo cuando el componente de vista de textura correspondiente se recibe correctamente. Incluir todos los elementos sintácticos para el componente de vista de profundidad puede hacer que una cabecera de fragmento para la unidad NAL del componente de vista de profundidad sea relativamente grande. El tamaño de la
55 cabecera de fragmento de profundidad se puede reducir mediante predicción de algunos elementos sintácticos a partir de elementos sintácticos de la cabecera de fragmento de textura para los componentes de vista de textura. Además, el tamaño de una cabecera de fragmento para una vista puede reducirse mediante predicción de algunos elementos sintácticos a partir de elementos sintácticos de otra vista.

60 Un flujo de bits se puede utilizar para transferir unidades de bloque de vídeo multivista más profundidad y elementos sintácticos entre, por ejemplo, el dispositivo de origen 12 y el dispositivo de destino 16 de la FIG. 1. El flujo de bits puede cumplir con la norma de codificación UIT H.264/AVC y, en particular, sigue una estructura de flujo de bits de codificación de vídeo multivista (MVC). Es decir, en algunos ejemplos, el flujo de bits se ajusta a la extensión MVC de la norma H.264/AVC. En otros ejemplos, el flujo de bits se ajusta a una extensión multivista de la HEVC o una
65 extensión multivista de otra norma. En otros ejemplos más, se utilizan otras normas de codificación.

Una disposición de orden (orden de decodificación) de flujo de bits MVC típica es una codificación centrada en el tiempo. Cada unidad de acceso se define para contener las imágenes codificadas de todas las vistas para un instante de tiempo de salida. El orden de decodificación de las unidades de acceso puede o no ser idéntico al orden de salida o de visualización. Típicamente, la predicción MVC puede incluir tanto la predicción inter de imágenes dentro de cada vista como la predicción inter de vistas. En la MVC, la predicción inter de vistas puede admitirse gracias a la compensación de movimiento de disparidad, que utiliza la sintaxis de la compensación de movimiento H.264/AVC, pero permite utilizar una imagen de una vista diferente como imagen de referencia.

La MVC admite la codificación de dos vistas. Una de las ventajas de la MVC es que un codificador MVC puede tomar más de dos vistas como entrada de vídeo 3D, y un decodificador MVC puede decodificar las dos vistas como una representación multivista. Por lo tanto, un renderizador con un decodificador MVC puede tratar un contenido de vídeo 3D como si tuviera varias vistas. Previamente, la MVC no procesaba la entrada de mapa de profundidad, de forma similar a la H.264/AVC con mensajes de información de mejora complementaria (SEI) (información estéreo o imágenes de intercalado espacial).

En la norma H.264/AVC, se definen unidades de capa de abstracción de red (NAL) para una representación de vídeo adecuada para la red destinada a aplicaciones como la videotelefonía, el almacenamiento o la transmisión en continuo de vídeo. Las unidades NAL pueden clasificarse en unidades NAL de capa de codificación de vídeo (VCL) y unidades NAL no VCL. Las unidades VCL pueden contener un motor de compresión de núcleo y comprender niveles de bloque, macrobloque (MB) y fragmento. Otras unidades NAL son unidades NAL no VCL.

En un ejemplo de codificación de vídeo 2D, cada unidad NAL contiene una cabecera de unidad NAL de un byte y una carga útil de tamaño variable. Se utilizan cinco bits para especificar el tipo de unidad NAL. Se utilizan tres bits para `nal_ref_idc`, que indica la importancia de la unidad NAL en términos de cómo otras imágenes (unidades NAL) se refieren a ella. Por ejemplo, si `nal_ref_idc` se iguala a 0 significa que la unidad NAL no se utiliza para la predicción inter. Puesto que la norma H.264/AVC se amplía para incluir la codificación de vídeo 3D, como la norma de codificación de vídeo escalable (SVC), la cabecera NAL puede ser similar a la del entorno 2D. Por ejemplo, uno o más bits de la cabecera de unidad NAL se utilizan para indicar que la unidad NAL es una unidad NAL de cuatro componentes.

Las cabeceras de unidad NAL también se pueden utilizar como unidades NAL MVC. Sin embargo, en la MVC, la estructura de cabecera de unidad NAL puede conservarse excepto para las unidades NAL de prefijo y las unidades NAL de fragmento codificadas mediante MVC. Las unidades NAL del fragmento codificadas mediante MVC pueden comprender una cabecera de cuatro bytes y la carga útil de la unidad NAL, que puede incluir una unidad de bloque tal como el bloque codificado 8 de la FIG. 1. Los elementos sintácticos de la cabecera de unidad NAL MVC pueden incluir `priority_id`, `temporal_id`, `anchor_pic_flag`, `view_id`, `non_idr_flag` e `inter_view_flag`. En otros ejemplos, se incluyen otros elementos sintácticos en una cabecera de unidad NAL MVC.

El elemento sintáctico `anchor_pic_flag` puede indicar si una imagen es una imagen anclada o no anclada. Las imágenes ancladas y todas las imágenes sucesivas en el orden de salida (es decir, el orden de visualización) se pueden decodificar correctamente sin decodificar las imágenes previas en el orden de decodificación (es decir, el orden de flujo de bits) y por lo tanto pueden utilizarse como puntos de acceso aleatorio. Las imágenes ancladas y las imágenes no ancladas pueden tener diferentes dependencias, ambas de las cuales pueden señalarse en el conjunto de parámetros de secuencia.

La estructura de flujo de bits definida en la MVC puede caracterizarse por dos elementos sintácticos: `view_id` y `temporal_id`. El elemento sintáctico `view_id` puede indicar el identificador de cada vista. Este identificador de la cabecera de la unidad NAL permite la fácil identificación de unidades NAL en el decodificador y el acceso rápido de las vistas decodificadas para su visualización. El elemento sintáctico `temporal_id` puede indicar la jerarquía de escalabilidad temporal o, indirectamente, la velocidad de trama. Por ejemplo, un punto de operación que incluye unidades NAL con un valor `temporal_id` máximo más pequeño puede tener una velocidad de trama más baja que un punto de operación con un valor `temporal_id` máximo mayor. Las imágenes codificadas con un valor `temporal_id` superior típicamente dependen de las imágenes codificadas con valores `temporal_id` inferiores dentro de una vista, pero pueden no depender de ninguna imagen codificada con un `temporal_id` superior.

Los elementos sintácticos `view_id` y `temporal_id` de la cabecera de unidad NAL pueden utilizarse para la extracción y adaptación de flujos de bit. El elemento sintáctico `priority_id` puede utilizarse principalmente para el proceso simple de adaptación de flujo de bits de una trayectoria. El elemento sintáctico `inter_view_flag` puede indicar si esta unidad NAL se utilizará para la predicción inter de vistas de otra unidad NAL en una vista diferente.

La MVC también puede emplear conjuntos de parámetros de secuencia (SPS) e incluir una extensión MVC SPS. Los conjuntos de parámetros se utilizan para la señalización en la norma H.264/AVC. Los conjuntos de parámetros de secuencia comprenden información de cabecera de nivel de secuencia. Los conjuntos de parámetros de imagen (PPS) comprenden la información de cabecera de nivel de imagen que cambia raramente. Con los conjuntos de parámetros, esta información que cambia raramente no siempre se repite para cada secuencia o imagen, de ahí que mejore la eficiencia de codificación. Además, el uso de conjuntos de parámetros permite la transmisión fuera de

banda de la información de cabecera, evitando la necesidad de transmisiones redundantes, para la recuperación frente a errores. En algunos ejemplos de transmisión fuera de banda, las unidades NAL de conjunto de parámetros se transmiten en un canal diferente al de las otras unidades NAL. En la MVC, una dependencia de vista se puede señalar en la extensión MVC SPS. Todas las predicciones inter de vistas pueden realizarse dentro del alcance especificado por la extensión MVC SPS.

En algunas técnicas de codificación de vídeo 3D previas, el contenido se codifica de tal manera que los componentes de color, por ejemplo, los del espacio de color YCbCr, están codificados en una o más unidades NAL, mientras que la imagen de profundidad está codificada en una o más unidades NAL separadas. Sin embargo, cuando ni una sola unidad NAL contiene las muestras codificadas de imágenes de textura y profundidad de una unidad de acceso, pueden producirse varios problemas. Por ejemplo, en un decodificador de vídeo 3D, se espera que, después de decodificar tanto la imagen de textura como de profundidad de cada trama, se active el renderizado de vistas basado en el mapa de profundidad y textura para generar las vistas virtuales. Si la unidad NAL de la imagen de profundidad y la unidad NAL de la textura para una unidad de acceso están codificadas de manera secuencial, el renderizado de vistas no puede comenzar hasta que se decodifique toda la unidad de acceso. Esto puede conducir a un aumento del tiempo que lleva renderizar el vídeo 3D.

Además, la imagen de textura y la imagen de mapa de profundidad asociada pueden compartir alguna información en varios niveles del códec, por ejemplo, nivel de secuencia, nivel de imagen, nivel de fragmento y nivel de bloque. Codificar esta información en dos unidades NAL puede crear una carga de implementación adicional al compartir o predecir la información. Por lo tanto, el codificador puede tener que realizar dos veces la estimación de movimiento para una trama, una para la textura y otra para el mapa de profundidad. De manera similar, el decodificador puede necesitar realizar dos veces una compensación de movimiento para una trama.

Como se describe en el presente documento, a unas normas existentes, tales como la MVC, se les añade unas técnicas con el fin de admitir vídeo 3D. Se puede añadir vídeo multivista más profundidad (MVD) a la MVC para el procesamiento de vídeo 3D. Las técnicas de codificación de vídeo 3D pueden incluir más flexibilidad y extensibilidad a las normas de vídeo existentes, por ejemplo, para cambiar el ángulo de visión sin problemas o ajustar la convergencia o percepción de profundidad hacia delante o hacia atrás, lo cual puede basarse en las especificaciones de los dispositivos o las preferencias del usuario, por ejemplo. Las normas de codificación también pueden ampliarse para utilizar mapas de profundidad para la generación de vistas virtuales en vídeo 3D.

Las siguientes tablas muestran ejemplos de elementos sintácticos para implementar las técnicas de la presente divulgación.

Tabla 2 - Extensión 3DV de cabecera de fragmento

	C	Descriptor
slice_header_3DV_extension() {		
if (pred_slice_header_depth_idc == 0)		
slice_header()		
else {		
if (pred_slice_header_depth_idc == 2)		
first_mb_in_slice	2	ue(v)
pic_parameter_set_id	2	ue(v)
ref_pic_list_inherit_flag	2	u(1)
ref_pic_inside_view_or_base_view_flag	2	u(1)
if (!ref_pic_list_inherit_flag) {		
if(slice_type == P slice_type == SP slice_type == B) {		
num_ref_idx_active_override_flag	2	u(1)
if(num_ref_idx_active_override_flag) {		
num_ref_idx_l0_active_minus1	2	ue(v)
if(slice_type == B)		
num_ref_idx_l1_active_minus1	2	ue(v)
}		
}		
}		
if(slice_type % 5 != 2 && slice_type % 5 != 4)		
ref_pic_list_exclusion_or_insertion(0, abs(NumRef0Delta))		

if(slice_type % 5 == 1)		
ref_pic_list_exclusion_or_insertion(1, abs(NumRef1Delta))		
ref_pic_list_3vc_modification()	2	
}		
slice_qp_delta	2	se(v)
}		
}		

5 El elemento sintáctico **ref_pic_list_inherit_flag** igual a 1 indica que los elementos sintácticos relacionados con la construcción de la lista de imágenes de referencia son los mismos que los de la cabecera de fragmento de referencia. El elemento sintáctico **ref_pic_list_inherit_flag** igual a 0 indica que los elementos sintácticos relacionados con la construcción de la lista de imágenes de referencia pueden no ser los mismos que los de la cabecera de fragmento de referencia y pueden estar presentes en esta cabecera de fragmento.

10 El elemento sintáctico **ref_pic_inside_view_or_base_view_flag** igual a 1 indica que la cabecera de fragmento de referencia es la cabecera de fragmento del componente de vista de textura o profundidad que ya ha estado presente en el componente de vista actual de la misma unidad de acceso. El elemento sintáctico **ref_pic_inside_view_or_base_view_flag** igual a 0 indica que la cabecera de fragmento de referencia es la cabecera de fragmento de la vista básica compatible con AVC (textura).

15 En otro ejemplo, el elemento sintáctico **ref_pic_inside_view_or_base_view_flag** igual a 1 puede indicar que la cabecera del fragmento de referencia es la cabecera de fragmento del componente de vista de profundidad de la vista básica. Cuando **ref_pic_inside_view_or_base_view_flag** es igual a 0, se introduce un indicador adicional para señalar si la cabecera de fragmento de referencia es la cabecera de fragmento de la textura o profundidad de la vista básica.

20 El elemento sintáctico **NumRefxDelta** se establece para ser igual a la diferencia entre el valor de num_ref_idx_lx_active_minus1 señalado en la cabecera de fragmento actual y el valor de num_ref_idx_lx_active_minus1 de la cabecera de fragmento de referencia.

Tabla 3 – Elemento sintáctico de exclusión o inserción de lista de imágenes de referencia

ref_pic_list_exclusion_or_insertion (x, N) {	C	Descriptor
ref_pic_list_insetion_lx	2	u(1)
if (!ref_pic_list_insetion_lx) {		
ref_pic_list_excluding_lx	2	u(1)
excluding_from_begining_or_ending_lx	2	u(1)
else {		
inserting_from_begining_or_ending_lx	2	u(1)
for (i =0; i < N; i ++) {		
modification_of_pic_nums_idc	2	ue(v)
if(modification_of_pic_nums_idc == 0 modification of pic nums idc == 1)		
Abs_diff_pic_num_minus1	2	ue(v)
else if(modification_of_pic_nums_idc == 2)		
long_term_pic_num	2	ue(v)
else if (modification_of_pic_nums_idc == 4 modification of pic nums idc == 5)		
abs_diff_view_idc_minus1	2	ue(v)
}		
}		
}		

25 El elemento sintáctico **ref_pic_list_insertion_lx** igual a 1 indica que las imágenes de referencia de RefPicListx se

añaden a la RefPicListx creada siguiendo los elementos sintácticos de la cabecera de fragmento de referencia. El elemento sintáctico **ref_pic_list_insertion_ix** igual a 0 indica que las imágenes de referencia de RefPicListx no se añaden a la RefPicListx creada siguiendo los elementos sintácticos de la cabecera de fragmento de referencia.

- 5 El elemento sintáctico **excluded_from_begining_or_ending_ix** igual a 1 indica que las imágenes de referencia de RefPicListx están excluidas de la RefPicListx creada siguiendo los elementos sintácticos de la cabecera de fragmento de referencia. El elemento sintáctico **excluded_from_begining_or_ending_ix** igual a 0 indica que la sintaxis de modificación de imágenes de referencia puede señalarse para RefPicListx. Por lo tanto, la construcción de la lista de imágenes de referencia para RefPicListx se basa en la sintaxis señalada en esta cabecera de fragmento. Cuando no está presente, se deduce que este indicador es igual a 0.

El elemento sintáctico NewRPLRFlagx se establece de la siguiente manera:

- ```
15 if (ref_pic_list_inserion_ix) NewRPLRFlagx = false;
 else if (ref_pic_list_excluding_ix) NewRPLRFlagx = false;
 else NewRPLRFlagx = true
```

- 20 El proceso de decodificación para la inserción de la lista de imágenes de referencia es similar a la modificación de la lista de imágenes de referencia (RPLM), con el número de partida picNumber establecido de la manera indicada a continuación. Si las imágenes de referencia se insertan al principio de una lista (generada por la sintaxis de la cabecera de fragmento de referencia), el picNumber se iguala al número de partida de una nueva RPLM. Si las imágenes de referencia se insertan al final de una lista, el picNumber se establece en el número de la trama reordenada en último lugar basándose en la RPLM.

**Tabla 4 - Elementos sintácticos de modificación de lista de imágenes de referencia**

|                                                                                  | <b>C</b> | <b>Descriptor</b> |
|----------------------------------------------------------------------------------|----------|-------------------|
| ref_pic_list_3vc_modification() {                                                |          |                   |
| if(slice_type % 5 != 2 && slice_type % 5 != 4 && ! NewRPLRFlag0) {               |          |                   |
| <b>ref_pic_list_modification_flag_I0</b>                                         | 2        | u(1)              |
| if(ref_pic_list_modification_flag_I0)                                            |          |                   |
| do {                                                                             |          |                   |
| <b>modification_of_pic_nums_idc</b>                                              | 2        | ue(v)             |
| if(modification_of_pic_nums_idc == 0    modification of pic nums idc == 1)       |          |                   |
| <b>abs_diff_pic_num_minus1</b>                                                   | 2        | ue(v)             |
| else if(modification_of_pic_nums_idc == 2)                                       |          |                   |
| <b>long_term_pic_num</b>                                                         | 2        | ue(v)             |
| else if (modification_of_pic_nums_idc == 4    modification of pic nums idc == 5) |          |                   |
| <b>abs_diff_view_idx_minus1</b>                                                  | 2        | ue(v)             |
| } while modification_of_pic_nums_idc != 3)                                       |          |                   |
| }                                                                                |          |                   |
| if(slice_type % 5 == 1 && ! NewRPLRFlag1)) {                                     |          |                   |
| <b>ref_pic_list_modification_flag_I1</b>                                         | 2        | u(1)              |
| if(ref_pic_list_modification_flag_I1)                                            |          |                   |
| do {                                                                             |          |                   |
| <b>modification_of_pic_nums_idc</b>                                              | 2        | ue(v)             |
| if(modification_of_pic_nums_idc == 0    modification of pic nums idc == 1)       |          |                   |
| <b>abs_diff_pic_num_minus1</b>                                                   | 2        | ue(v)             |
| else if(modification_of_pic_nums_idc == 2)                                       |          |                   |
| <b>long_term_pic_num</b>                                                         | 2        | ue(v)             |
| else if (modification_of_pic_nums_idc == 4    modification of pic nums idc == 5) |          |                   |

|                                            |   |       |
|--------------------------------------------|---|-------|
| <b>abs_diff_view_idx_minus1</b>            | 2 | ue(v) |
| } while(modification_of_pic_nums_idc != 3) |   |       |
| }                                          |   |       |
| }                                          |   |       |

La Tabla 5 ilustra otro ejemplo de extensión 3DV de cabecera de fragmento.

**TABLA 5 - Extensión 3DV de cabecera de fragmento**

|                                                                     | <b>C</b> | <b>Descriptor</b> |
|---------------------------------------------------------------------|----------|-------------------|
| slice_header() {                                                    |          |                   |
| <b>first_mb_in_slice</b>                                            | 2        | ue(v)             |
| <b>slice_type</b>                                                   | 2        | ue(v)             |
| <b>pic_parameter_set_id</b>                                         | 2        | ue(v)             |
| <b>slice_id</b>                                                     | 2        | ue(v)             |
| if(slice_header_prediction_idc != 0) {                              |          |                   |
| <b>ref_pic_list_inherit_flag</b>                                    | 2        | u(1)              |
| <b>delta view idx</b>                                               | 2        | ue(v)             |
| if (delta_view_idx >0)                                              |          |                   |
| <b>from_texture_depth_flag</b>                                      | 2        | u(1)              |
| <b>slice_id_in_ref_view_component</b>                               | 2        | ue(v)             |
| if (! ref_pic_list_inherit_flag) {                                  |          |                   |
| if(slice_type == P    slice_type == SP<br>   slice type == B) {     |          |                   |
| <b>num_ref_idx_active_override_flag</b>                             | 2        | u(1)              |
| if(num_ref_idx_active_override_flag) {                              |          |                   |
| <b>num_ref_idx_l0_active_minus1</b>                                 | 2        | ue(v)             |
| if(slice_type == B)                                                 |          |                   |
| <b>num_ref_idx_l1_active_minus1</b>                                 | 2        | ue(v)             |
| }                                                                   |          |                   |
| }                                                                   |          |                   |
| ref_pic_list_mvc_modification()                                     | 2        |                   |
| }                                                                   |          |                   |
| <b>slice_qp_delta</b>                                               | 2        | se(v)             |
| } else {                                                            |          |                   |
| if(separate_colour_plane_flag == 1)                                 |          |                   |
| <b>colour_plane_id</b>                                              | 2        | u(2)              |
| <b>frame_num</b>                                                    | 2        | u(v)              |
| if(!frame_mbs_only_flag) {                                          |          |                   |
| <b>field_pic_flag</b>                                               | 2        | u(1)              |
| if(field_pic_flag)                                                  |          |                   |
| <b>bottom_field_flag</b>                                            | 2        | u(1)              |
| }                                                                   |          |                   |
| if(!idrPicFlag)                                                     |          |                   |
| <b>idr_pic_id</b>                                                   | 2        | ue(v)             |
| if(pic_order_cnt_type == 0) {                                       |          |                   |
| <b>pic_order_cnt_lsb</b>                                            | 2        | u(v)              |
| if(bottom_field_pic_order_in_frame_present_flag && !field pic flag) |          |                   |



|                                                                                                                    |   |       |
|--------------------------------------------------------------------------------------------------------------------|---|-------|
| <b>delta_pic_order_cnt_bottom</b>                                                                                  | 2 | se(v) |
| }                                                                                                                  |   |       |
| if(pic_order_cnt_type == 1 && !delta_pic_order_always_zero_flag) {                                                 |   |       |
| <b>delta_pic_order_cnt[0]</b>                                                                                      | 2 | se(v) |
| if(bottom_field_pic_order_in_frame_present_flag && !field_pic_flag)                                                |   |       |
| <b>delta_pic_order_cnt[1]</b>                                                                                      | 2 | se(v) |
| }                                                                                                                  |   |       |
| if(redundant_pic_cnt_present_flag)                                                                                 |   |       |
| <b>redundant_pic_cnt</b>                                                                                           | 2 | ue(v) |
| if(slice_type == B)                                                                                                |   |       |
| <b>direct_spatial_mv_pred_flag</b>                                                                                 | 2 | u(1)  |
| if(slice_type == P    slice_type == SP    slice_type == B) {                                                       |   |       |
| <b>num_ref_idx_active_override_flag</b>                                                                            | 2 | u(1)  |
| if(num_ref_idx_active_override_flag) {                                                                             |   |       |
| <b>num_ref_idx_l0_active_minus1</b>                                                                                | 2 | ue(v) |
| if(slice_type == B)                                                                                                |   |       |
| <b>num_ref_idx_l1_active_minus1</b>                                                                                | 2 | ue(v) |
| }                                                                                                                  |   |       |
| }                                                                                                                  |   |       |
| if(nal_unit_type == 20    (nal_unit_type == 21 && DepthFlag))                                                      |   |       |
| ref_pic_list_mvc_modification() /* specified in Annex H */                                                         | 2 |       |
| else if(nal_unit_type == 21 && DepthFlag == 0)                                                                     |   |       |
| /* it has not been decided yet, which reference picture list modification syntax and decoding process is used */   |   |       |
| else                                                                                                               |   |       |
| ref_pic_list_modification()                                                                                        | 2 |       |
| if((weighted_pred_flag && (slice_type == P    slice_type == SP))    (weighted_bipred_idc == 1 && slice_type == B)) |   |       |
| pred_weight_table()                                                                                                | 2 |       |
| if(nal_ref_idc != 0)                                                                                               |   |       |
| dec_ref_pic_marking()                                                                                              | 2 |       |
| if(entropy_coding_mode_flag && slice_type != I && slice_type != SI)                                                |   |       |
| <b>cabac_init_idc</b>                                                                                              | 2 | ue(v) |
| <b>slice_qp_delta</b>                                                                                              | 2 | se(v) |
| if(slice_type == SP    slice_type == SI) {                                                                         |   |       |
| if(slice_type == SP)                                                                                               |   |       |
| <b>sp_for_switch_flag</b>                                                                                          | 2 | u(1)  |
| <b>slice_qs_delta</b>                                                                                              | 2 | se(v) |
| }                                                                                                                  |   |       |
| if(deblocking_filter_control_present_flag) {                                                                       |   |       |
| <b>disable_deblocking_filter_idc</b>                                                                               | 2 | ue(v) |
| if(disable_deblocking_filter_idc != 1) {                                                                           |   |       |
| <b>slice_alpha_c0_offset_div2</b>                                                                                  | 2 | se(v) |
| <b>slice_beta_offset_div2</b>                                                                                      | 2 | se(v) |
| }                                                                                                                  |   |       |

|                                                                                              |   |      |
|----------------------------------------------------------------------------------------------|---|------|
| }                                                                                            |   |      |
| if(num_slice_groups_minus1 > 0 &&<br>slice_group_map_type >= 3 && slice_group_map_type <= 5) |   |      |
| <b>slice_group_change_cycle</b>                                                              | 2 | u(v) |
| if(nal_unit_type == 21 && (slice_type != I && slice_type != SI)) {                           |   |      |
| if(DepthFlag)                                                                                |   |      |
| <b>depth_weighted_pred_flag</b>                                                              | 2 | u(1) |
| else                                                                                         |   |      |
| <b>dmvp_flag</b>                                                                             | 2 | u(1) |
| }                                                                                            |   |      |
| }                                                                                            |   |      |
| }                                                                                            |   |      |

Debe tenerse en cuenta que cada componente de vista puede estar predefinido o señalado en el conjunto de parámetros de secuencia. Es decir, cada componente de vista puede estar predefinido o señalado en el SPS, independientemente de si es un componente de vista de textura o un componente de vista de profundidad.

5 El elemento sintáctico **delta\_view\_idx** especifica a partir de qué índice de orden de vista del componente de vista se predice la cabecera de fragmento actual. VOIdx (índice de orden de vista) de la vista actual menos delta\_view\_idx es el índice de orden de vista del componente de vista de referencia. Cuando delta\_view\_idx es igual a 0, el componente de vista de referencia es el componente de vídeo codificado anterior de la misma vista de la misma unidad de acceso. Si un componente de vista de referencia está en una vista diferente (tal como la vista A), la vista diferente (vista A) debe ser una vista dependiente de la vista actual. La vista diferente (vista A) debe ser una vista directa o indirectamente dependiente de la vista actual.

15 El indicador **from\_texture\_depth\_flag** puede incluirse en la cabecera de fragmento. Cuando **from\_texture\_depth\_flag** es igual a 1, se indica que el componente de vista de referencia es un componente de vista de textura. Cuando **from\_texture\_depth\_flag** es igual a 0, se indica que el componente de vista de referencia es un componente de vista de profundidad.

20 El elemento sintáctico **slice\_id\_in\_ref\_view\_component** especifica el ID del fragmento en el componente de vista de referencia. Cuando se predice una cabecera de fragmento a partir de un componente de vista que pertenece a la textura de la vista básica, **slice\_id\_in\_ref\_view\_component** será igual a 0.

25 El elemento sintáctico **slice\_id** especifica un identificador del fragmento actual en el componente de vista actual. El valor de **slice\_id** puede ser igual a 0 para el fragmento que incluye el bloque de codificación superior izquierdo. El valor de **slice\_id** puede incrementarse en uno por cada fragmento en el orden de decodificación.

De forma alternativa, en algunos ejemplos, la condición anterior «if (delta\_view\_idx > 0)» se puede eliminar de la tabla de sintaxis. En ese caso, siempre se puede señalar **from\_texture\_depth\_flag**.

30 De forma alternativa, el elemento sintáctico **slice\_id** puede señalarse además para cualquier componente de vista que no sea la textura de la vista básica.

35 De forma alternativa, puede ser solo necesario diferenciar el valor de **slice\_id** para cada fragmento de un componente de vista. En dicho ejemplo, **slice\_id** puede no necesitar ser 0 para el fragmento que incluye el bloque de codificación superior izquierdo.

40 A continuación se describe el proceso general de codificación de vídeo de acuerdo con la norma H.264/AVC. Una secuencia de vídeo incluye típicamente una serie de tramas de vídeo. Un grupo de imágenes (GOP) comprende en general una serie de una o más tramas de vídeo. Un GOP puede incluir datos sintácticos en una cabecera del GOP, en una cabecera de una o más tramas del GOP o en otras ubicaciones, que indican el número de tramas incluidas en el GOP. Cada trama puede incluir datos sintácticos de trama que describen una modalidad de codificación para la respectiva trama. Los codificadores y decodificadores de vídeo actúan típicamente sobre bloques de vídeo de tramas de vídeo individuales, con el fin de codificar y/o decodificar los datos de vídeo. Un bloque de vídeo puede corresponder a un macrobloque o una división de un macrobloque. Los bloques de vídeo pueden presentar tamaños

45 fijos o variables y pueden diferir en tamaño de acuerdo con una norma de codificación especificada. Cada trama de vídeo puede incluir una pluralidad de fragmentos. Cada fragmento puede incluir una pluralidad de macrobloques, que pueden disponerse en divisiones, también denominadas subbloques.

En un ejemplo, la norma UIT-T H.264 admite la predicción intra en varios 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 predicción inter en varios tamaños de bloque, tales como 16x16, 16x8, 8x16, 8x8, 8x4, 4x8 y 4x4 para componentes de luma y tamaños escalados correspondientes para componentes de croma. En la presente divulgación, «NxN» y «N por N» pueden utilizarse indistintamente para hacer referencia a las dimensiones de píxel del bloque en términos de 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 la dirección horizontal ( $x = 16$ ). Asimismo, un bloque NxN presenta 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 pueden estar dispuestos en filas y columnas. Además, los bloques no necesitan presentar necesariamente el mismo número de píxeles en la dirección horizontal y en la dirección vertical. Por ejemplo, los bloques pueden comprender NxM píxeles, donde M no es necesariamente igual a N.

Los tamaños de bloque que son inferiores a 16 por 16 pueden denominarse divisiones de un macrobloque de 16 por 16. Los bloques de vídeo pueden comprender bloques de datos de píxel en el dominio del píxel, o bloques de coeficientes de transformada en el dominio de la transformada, por ejemplo, tras la aplicación de una transformada, tal como una transformada discreta del coseno (DCT), una transformada entera, una transformada de ondícula o una transformada conceptualmente similar, a los datos de bloque de vídeo residual que representan diferencias de píxel entre bloques de vídeo codificados y bloques de vídeo predictivos. En algunos casos, un bloque de vídeo puede comprender bloques de coeficientes de transformada cuantificados en el dominio de transformada.

Los bloques de vídeo más pequeños pueden tener una mejor resolución y pueden utilizarse en ubicaciones de una trama de vídeo que incluyen altos niveles de detalle. En general, los macrobloques y las diversas divisiones, denominadas en ocasiones subbloques, pueden considerarse bloques de vídeo. Además, un fragmento puede considerarse una pluralidad de bloques de vídeo, tales como macrobloques y/o subbloques. Cada fragmento puede ser una unidad independientemente decodificable de una trama de vídeo. De forma alternativa, las propias tramas pueden ser unidades decodificables, o pueden definirse otras partes de una trama como unidades decodificables. El término «unidad codificada» puede referirse a cualquier unidad independientemente decodificable de una trama de vídeo, tal como una trama completa, un fragmento de una trama, un grupo de imágenes (GOP), denominado también secuencia, u otra unidad independientemente decodificable definida de acuerdo con unas técnicas de codificación aplicables.

Tras una codificación de predicción intra o predicción inter para generar datos predictivos y datos residuales, y tras cualquier transformada (tal como la transformada entera 4x4 u 8x8 utilizada en la norma H.264/AVC o una transformada discreta del coseno DCT) aplicada a datos residuales para generar coeficientes de transformada, puede llevarse a cabo la cuantificación de los coeficientes de transformada. La cuantificación se refiere en general a un proceso en el que los coeficientes de transformada se cuantifican para reducir posiblemente la cantidad de datos utilizados para representar los coeficientes. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos o todos los coeficientes. Por ejemplo, un valor de  $n$  bits puede redondearse a la baja a un valor de  $m$  bits durante la cuantificación, donde  $n$  es mayor que  $m$ .

Tras la cuantificación puede llevarse a cabo la codificación de entropía de los datos cuantificados, por ejemplo de acuerdo con la codificación de longitud variable adaptativa al contenido (CAVLC), la codificación aritmética binaria adaptativa al contexto (CABAC), la codificación de entropía mediante división en intervalos de probabilidad (PIPE) u otra metodología de codificación de entropía. Una unidad de procesamiento configurada para la codificación de entropía, u otra unidad de procesamiento, puede desempeñar otras funciones de procesamiento, tales como la codificación de coordenada diferencial cero de coeficientes cuantificados y/o la generación de información de sintaxis, tal como valores de patrón de bloque codificado (CBP), tipo de macrobloque, modalidad de codificación, tamaño máximo de macrobloque para una unidad codificada (tal como una trama, un fragmento, un macrobloque o una secuencia) o similar.

El codificador de vídeo puede generar además datos sintácticos, tales como datos sintácticos de bloque, datos sintácticos de trama y datos sintácticos de GOP, para un decodificador de vídeo, por ejemplo, en una cabecera de trama, una cabecera de bloque, una cabecera de fragmento o una cabecera de GOP. Los datos sintácticos de GOP pueden describir un número de tramas en el respectivo GOP, y los datos sintácticos de trama pueden indicar una modalidad de codificación/predicción utilizada para codificar la trama correspondiente. El decodificador de vídeo puede interpretar y utilizar los datos sintácticos en el proceso de decodificación.

En H.264/AVC, los bits de vídeo codificados se organizan en unidades de capa de abstracción de red (NAL), que proporcionan una representación de vídeo adecuada para la red destinada a aplicaciones tales como la videotelefonía, el almacenamiento, la radiodifusión o la transmisión en continuo. Las unidades NAL pueden clasificarse en unidades NAL de capa de codificación de vídeo (VCL) y unidades NAL no VCL. Las unidades VCL contienen el motor de compresión del núcleo y comprenden los niveles de bloque, MB y fragmento. Otras unidades NAL son unidades NAL no VCL.

Cada unidad NAL contiene una cabecera de unidad NAL de 1 byte. Se utilizan cinco bits para especificar el tipo de unidad NAL y se utilizan tres bits para `nal_ref_idc`, que indica la importancia de la unidad NAL en términos de cómo

otras imágenes (unidades NAL) se refieren a ella. Este valor igual a 0 significa que la unidad NAL no se utiliza para la predicción inter.

5 Los conjuntos de parámetros contienen información de cabecera de nivel de secuencia en conjuntos de parámetros de secuencia (SPS), e información de cabecera de nivel de imagen que cambia raramente en conjuntos de parámetros de imagen (PPS). Con los conjuntos de parámetros, esta información que cambia raramente no necesita repetirse para cada secuencia o imagen, por lo que se mejora la eficiencia de codificación. Además, el uso de conjuntos de parámetros permite la transmisión fuera de banda de la información de cabecera, evitando la necesidad de transmisiones redundantes para la recuperación frente a errores. En la transmisión fuera de banda, las unidades NAL de conjuntos de parámetros se pueden transmitir por un canal diferente al de otras unidades NAL.

15 En MVC, la predicción inter de vistas es admitida gracias a la compensación de disparidad, que utiliza la sintaxis de la compensación de movimiento H.264/AVC, pero permite utilizar una imagen de una vista diferente como imagen de referencia. Es decir, las imágenes en MVC pueden ser sometidas a predicción inter y codificación. Pueden utilizarse vectores de disparidad para la predicción inter de vistas, de una manera similar a los vectores de movimiento en la predicción temporal. Sin embargo, en lugar de proporcionar una indicación de movimiento, los vectores de disparidad indican el desplazamiento de datos en un bloque predicho con respecto a una trama de referencia de una vista diferente, para tener en cuenta el desplazamiento horizontal de la perspectiva de cámara de la escena común. De esta manera, una unidad de compensación de movimiento puede realizar una compensación de disparidad para la predicción inter de vistas.

20 Como se ha mencionado anteriormente, en la norma H.264/AVC, una unidad NAL incluye una cabecera de 1 byte y una carga útil de tamaños variables. En MVC, esta estructura se conserva excepto en las unidades NAL de prefijo y las unidades NAL de fragmento codificadas mediante MVC, que incluyen una cabecera de 4 bytes y la carga útil de la unidad NAL. Los elementos sintácticos de la cabecera de la unidad NAL MVC incluyen *priority\_id*, *temporal\_id*, *anchor\_pic\_flag*, *view\_id*, *non\_idr\_flag* e *inter\_view\_flag*.

30 El elemento sintáctico *anchor\_pic\_flag* indica si una imagen es una imagen anclada o no anclada. Las imágenes ancladas y todas las imágenes que suceden a estas en el orden de salida (es decir, el orden de visualización) se pueden decodificar correctamente sin decodificar las imágenes anteriores en el orden de decodificación (es decir, el orden de flujo de bits), y por lo tanto pueden utilizarse como puntos de acceso aleatorio. Las imágenes ancladas y las no ancladas pueden tener dependencias diferentes, ambas de las cuales están señalizadas en el conjunto de parámetros de secuencia.

35 La estructura del flujo de bits definida en MVC se caracteriza por dos elementos sintácticos: *view\_id* y *temporal\_id*. El elemento sintáctico *view\_id* indica el identificador de cada vista. Esta indicación en la cabecera de la unidad NAL permite la fácil identificación de las unidades NAL en el decodificador y el acceso rápido a las vistas decodificadas para su visualización. El elemento sintáctico *temporal\_id* indica la jerarquía de escalabilidad temporal o, indirectamente, la velocidad de trama. Un punto de operación que incluye unidades NAL con un valor *temporal\_id* máximo más pequeño tiene una velocidad de trama más baja que un punto de operación con un valor *temporal\_id* máximo mayor. Las imágenes codificadas con un valor *temporal\_id* superior dependen típicamente de las imágenes codificadas con valores *temporal\_id* inferiores dentro de una vista, pero no de ninguna imagen codificada con un valor *temporal\_id* superior.

45 Los elementos sintácticos *view\_id* y *temporal\_id* de la cabecera de la unidad NAL se utilizan tanto para la extracción como para la adaptación del flujo de bits. Otro elemento sintáctico de la cabecera de la unidad NAL es *priority\_id*, que se utiliza para el proceso simple de adaptación de flujo de bits de una trayectoria. Es decir, un dispositivo que recibe o recupera el flujo de bits puede utilizar el valor *priority\_id* para determinar las prioridades entre las unidades NAL al preprocesar la extracción y la adaptación del flujo de bits, lo que permite enviar un flujo de bits a varios dispositivos de destino con diversas capacidades de codificación y renderizado.

50 El elemento sintáctico *inter\_view\_flag* indica si la unidad NAL se utilizará para la predicción inter de vistas de otra unidad NAL en una vista diferente.

55 En MVC, la dependencia de la vista se señala en la extensión MVC del SPS. Toda la predicción inter de vistas se realiza dentro del alcance especificado por la extensión MVC del SPS. La dependencia de la vista indica si una vista depende de otra vista, por ejemplo, para la predicción inter de vistas. Cuando se predice una primera vista a partir de datos de una segunda vista, se dice que la primera vista depende de la segunda vista. La Tabla 6 siguiente representa un ejemplo de extensión MVC para el SPS.

60

**TABLA 6**

|                                        |   |            |
|----------------------------------------|---|------------|
| seq_parameter_set_mvc_extension() {    | C | Descriptor |
| num_views_minus 1                      | 0 | ue(v)      |
| for(i = 0; i <= num_views_minus1; i++) |   |            |

|                                                                   |   |       |
|-------------------------------------------------------------------|---|-------|
| view id[i]                                                        | 0 | ue(v) |
| for(i = 1; i <= num_views_minus1; i++) {                          |   |       |
| num_anchor_refs_10[i]                                             | 0 | ue(v) |
| for(j = 0; j < num_anchor_refs_10[i]; j++)                        |   |       |
| anchor_ref_10[i][j]                                               | 0 | ue(v) |
| num_anchor_refs_l1[i]                                             | 0 | ue(v) |
| for(j = 0; j < num_anchor_refs_l1[i]; j++)                        |   |       |
| anchor_ref_l1[i][j]                                               | 0 | ue(v) |
| }                                                                 |   |       |
| for(i = 1; i <= num_views_minus_1; i++) {                         |   |       |
| num_non_anchor_refs_10[i]                                         | 0 | ue(v) |
| for(j = 0; j < num_non_anchor_refs_10[i]; j++)                    |   |       |
| non_anchor_ref_10[i][j]                                           | 0 | ue(v) |
| num_non_anchor_refs_l1[i]                                         | 0 | ue(v) |
| for(j = 0; j < num_non_anchor_refs_l1[i]; j++)                    |   |       |
| non_anchor_ref_l1[i][j]                                           | 0 | ue(v) |
| }                                                                 |   |       |
| num_level_values_signalled_minus_1                                | 0 | ue(v) |
| for(i = 0; i <= num_level_values_signalled_minus_1; i++) {        |   |       |
| level_idc[i]                                                      | 0 | u(8)  |
| num_applicable_ops_minus1[i]                                      | 0 | ue(v) |
| for(j = 0; j <= num_applicable_ops_minus1[i]; j++) {              |   |       |
| applicable_op_temporal_id[i][j]                                   | 0 | u(3)  |
| applicable_op_num_target_views_minus1[i][j]                       | 0 | ue(v) |
| for(k = 0; k <= applicable_op_num_target_views_minus1[i][j]; k++) |   |       |
| applicable_op_target_view_id[i][j][k]                             | 0 | ue(v) |
| applicable_op_num_views_minus1[i][j]                              | 0 | ue(v) |
| }                                                                 |   |       |
| }                                                                 |   |       |
| }                                                                 |   |       |

La FIG. 5 es un diagrama de bloques que ilustra un ejemplo del decodificador 28 de la FIG. 1 con más detalle, de acuerdo con las técnicas de la presente divulgación. El decodificador de vídeo 28 es un ejemplo de dispositivo o aparato informático de vídeo especializado, denominado «codificador» en el presente documento. Como se muestra en la FIG. 5, el decodificador de vídeo 28 corresponde al decodificador de vídeo 28 del dispositivo de destino 16. Sin embargo, en otros ejemplos, el decodificador de vídeo 28 corresponde a un dispositivo diferente. En otros ejemplos, otras unidades (tales como, por ejemplo, otros codificadores/decodificadores (CÓDECS)) también pueden realizar técnicas similares a las del decodificador de vídeo 28.

- 5
- 10 El decodificador de vídeo 28 incluye una unidad de decodificación de entropía 52 que realiza la decodificación de entropía del flujo de bits recibido para generar coeficientes cuantificados y los elementos sintácticos de predicción. El flujo de bits incluye bloques codificados que tienen componentes de textura y un componente de profundidad para cada ubicación de píxel, con el fin de renderizar vídeo 3D y elementos sintácticos. Los elementos sintácticos de predicción incluyen al menos uno de entre una modalidad de codificación, uno o más vectores de movimiento, información que identifica una técnica de interpolación utilizada, coeficientes para utilizar en el filtrado de interpolación y otra información asociada con la generación del bloque de predicción.
- 15

- 20 Los elementos sintácticos de predicción, por ejemplo, los coeficientes, se transmiten a la unidad de procesamiento de predicción 55. La unidad de procesamiento de predicción 55 incluye un módulo de predicción de sintaxis de profundidad 66. Si se utiliza la predicción para codificar los coeficientes en relación con los coeficientes de un filtro fijo, o en relación unos con otros, la unidad de procesamiento de predicción 55 decodifica los elementos sintácticos para definir los coeficientes reales. El módulo de predicción de sintaxis de profundidad 66 predice elementos

sintácticos de profundidad para los componentes de vista de profundidad, a partir de elementos sintácticos de textura para los componentes de vista de textura.

5 Si la cuantificación se aplica a cualquiera de los elementos sintácticos de predicción, la unidad de cuantificación inversa 56 elimina dicha cuantificación. La unidad de cuantificación inversa 56 puede tratar de manera diferente los componentes de profundidad y textura para cada ubicación de píxel de los bloques codificados en el tren de bits codificado. Por ejemplo, cuando el componente de profundidad se ha cuantificado de manera diferente que los componentes de textura, la unidad de cuantificación inversa 56 procesa los componentes de profundidad y textura por separado. Los coeficientes de filtro, por ejemplo, pueden codificarse mediante predicción y cuantificarse de acuerdo con la presente divulgación, y en este caso, el decodificador de vídeo 28 utiliza la unidad de cuantificación inversa 56 para decodificar mediante predicción y decuantificar dichos coeficientes.

15 La unidad de procesamiento de predicción 55 genera datos de predicción basándose en los elementos sintácticos de predicción y uno o más bloques previamente decodificados que se almacenan en la memoria 62, de la misma manera que se ha descrito con detalle anteriormente con respecto a la unidad de procesamiento de predicción 32 del codificador de vídeo 22. En particular, la unidad de procesamiento de predicción 55 realiza una o más de las técnicas de vídeo multivista más profundidad de la presente divulgación, durante la compensación de movimiento, para generar un bloque de predicción que incorpora componentes de profundidad, así como componentes de textura. El bloque de predicción (así como un bloque codificado) puede tener una resolución diferente para los componentes de profundidad y para los componentes de textura. Por ejemplo, los componentes de profundidad tienen una precisión de un cuarto de píxel, mientras que los componentes de textura tienen una precisión de píxel entero. Así pues, el decodificador de vídeo 28 utiliza una o más de las técnicas de la presente divulgación en la generación de un bloque de predicción. En algunos ejemplos, la unidad de procesamiento de predicción 55 puede incluir una unidad de compensación de movimiento que comprende filtros utilizados para técnicas de filtración de interpolación y similares a la interpolación según la presente divulgación. El componente de compensación de movimiento no se muestra en la FIG. 5 para simplificar y facilitar la ilustración.

30 La unidad de cuantificación inversa 56 realiza la cuantificación inversa, es decir, decuantifica, los coeficientes cuantificados. El proceso de cuantificación inversa es un proceso definido para la decodificación H.264 o para cualquier otra norma de decodificación. La unidad de procesamiento de transformada inversa 58 aplica una transformada inversa, por ejemplo una DCT inversa o un proceso de transformada inversa conceptualmente similar, a los coeficientes de transformada, con el fin de generar bloques residuales en el dominio del píxel. El sumador 64 suma el bloque residual con el correspondiente bloque de predicción generado por la unidad de procesamiento de predicción 55 para formar una versión reconstruida del bloque original codificado por el codificador de vídeo 22. Si se desea, también se aplica un filtro de eliminación de bloques para filtrar los bloques decodificados, con el fin de eliminar distorsiones de efecto pixelado. Los bloques de vídeo decodificados se almacenan a continuación en la memoria 62, que incluye bloques de referencia para la compensación de movimiento posterior y también proporciona vídeo decodificado para activar el dispositivo de visualización (tal como el dispositivo de visualización 30 de la FIG. 1).

40 El vídeo decodificado se puede utilizar para renderizar vídeo 3D. El vídeo 3D puede comprender una vista virtual tridimensional. La información de profundidad se utiliza para determinar un desplazamiento horizontal (disparidad horizontal) para cada píxel del bloque. También se puede realizar la manipulación de oclusiones para generar la vista virtual. Los elementos sintácticos para los componentes de vista de profundidad se pueden predecir a partir de elementos sintácticos para los componentes de vista de textura.

50 La FIG. 6 es un diagrama de flujo que ilustra un ejemplo de operación de un codificador de vídeo, de acuerdo con las técnicas de la presente divulgación. En algunos ejemplos, el codificador de vídeo es un codificador de vídeo, tal como el codificador de vídeo 22 mostrado en las FIGS. 1 y 2. En otros ejemplos, el codificador de vídeo es un decodificador de vídeo, tal como el decodificador de vídeo 28 mostrado en las FIGS. 1 y 5.

55 El ejemplo de procedimiento de codificación de datos de vídeo incluye codificar uno o más bloques de datos de vídeo representativos de información de textura de al menos una parte de una trama de los datos de vídeo (102). Por ejemplo, el decodificador de vídeo 28 recibe un bloque codificado de datos de vídeo. En otro ejemplo, el codificador de vídeo 22 recibe un bloque codificado de datos de vídeo.

60 El ejemplo incluye además procesar un fragmento de textura para un componente de vista de textura de una vista actual asociada con una unidad de acceso, comprendiendo el fragmento de textura el uno o más bloques codificados y una cabecera de fragmento de textura que comprende un conjunto de elementos sintácticos representativos de características del fragmento de textura (104). Por ejemplo, el decodificador de vídeo 28 recibe el fragmento de textura codificado. En otro ejemplo, el codificador de vídeo 22 genera el fragmento de textura.

65 El ejemplo de procedimiento incluye además codificar información de profundidad representativa de los valores de profundidad para al menos la parte de la trama (106). El procedimiento también incluye procesar un fragmento de profundidad para un componente de vista de profundidad correspondiente al componente de vista de textura de la vista, comprendiendo el fragmento de profundidad la información de profundidad codificada y una cabecera de

fragmento de profundidad que comprende un conjunto de elementos sintácticos representativos de características del fragmento de profundidad (108). El procesamiento del fragmento de textura o el fragmento de profundidad puede incluir predecir al menos un elemento sintáctico de al menos uno del conjunto de elementos sintácticos representativos de características del fragmento de textura o el conjunto de elementos sintácticos representativos de características del fragmento de profundidad a partir de una cabecera de fragmento de referencia, respectivamente, en el que la cabecera de fragmento de referencia es de un componente de vista de la misma unidad de acceso.

En algunos ejemplos, la codificación de la vista actual basada en los elementos sintácticos predichos comprende codificar la vista actual. En otros ejemplos, la codificación de la vista actual basada en los elementos sintácticos predichos comprende decodificar la vista actual.

La cabecera de fragmento de referencia puede ser uno de entre el componente de vista de textura o el componente de vista de profundidad. La cabecera de fragmento de referencia puede comprender una vista de textura básica compatible con la codificación de vídeo avanzada (AVC) de la misma unidad de acceso. El conjunto de elementos sintácticos representativos de características del fragmento de textura y el conjunto de elementos sintácticos representativos de características del fragmento de profundidad se hallan en un orden diferente a un orden de cabecera de fragmento de AVC. Por ejemplo, la cabecera de fragmento de AVC puede tener un orden específico típico para los elementos sintácticos.

En algunos ejemplos, se señala un índice que indica la cabecera de fragmento de referencia. El índice puede señalarse como un orden de vista delta, en el que el orden de vista delta indica la ubicación de la cabecera de fragmento de referencia en relación con la vista actual. La cabecera de fragmento de referencia puede estar asociada con otra vista en una misma unidad de acceso que la vista actual. El procedimiento puede incluir además señalar el elemento sintáctico en una extensión de cabecera de fragmento.

Se puede señalar un indicador que indica si la cabecera de fragmento de referencia corresponde al componente de vista de profundidad o al componente de vista de textura. También se puede señalar, en la cabecera del fragmento, un ID de fragmento del fragmento de profundidad o el fragmento de textura del componente de vista de profundidad o textura indicado, respectivamente. El procedimiento también puede incluir señalar un ID de fragmento para el fragmento de profundidad y el fragmento de textura, en el que el ID de fragmento se señala en al menos una de entre la cabecera de fragmento o una cabecera de unidad de capa de abstracción de red (NAL) de una unidad NAL del componente de vista. En otro ejemplo, el procedimiento también incluye señalar un ID de fragmento para el fragmento de profundidad y el fragmento de textura, en el que el ID de fragmento se señala en una unidad de acceso que incluye una unidad de capa de abstracción de red (NAL) del componente de vista de profundidad y el componente de vista de textura, respectivamente. Como se describe en el presente documento, la señalización se realiza en un flujo de bits codificado.

Por ejemplo, el decodificador de vídeo 28 recibe un ID de fragmento. El decodificador de vídeo 28 recibe un índice que indica la cabecera de fragmento de referencia. En algunos ejemplos, el índice comprende un orden de vista delta, en el que el orden de vista delta indica la ubicación de la cabecera de fragmento de referencia con respecto a la vista actual, y en el que la cabecera de fragmento de referencia está asociado con otra vista en una misma unidad de acceso que la vista actual. El decodificador de vídeo 28 recibe un indicador que indica si la cabecera de fragmento de referencia corresponde al componente de vista de profundidad o al componente de vista de textura. En otro ejemplo, el decodificador de vídeo 28 recibe un ID de fragmento del fragmento de profundidad o el fragmento de textura del componente de vista de profundidad o textura indicado, respectivamente. El decodificador de vídeo 28 recibe un ID de fragmento para el fragmento de profundidad y el fragmento de textura, en el que el ID de fragmento se señala en al menos una de entre la cabecera de fragmento o una cabecera de unidad de capa de abstracción de red (NAL) de una unidad NAL del componente de vista. El decodificador de vídeo 28 recibe un ID de fragmento para el fragmento de profundidad y el fragmento de textura, en el que el ID de fragmento se señala en una unidad de acceso que incluye una unidad de capa de abstracción de red (NAL) del componente de vista de profundidad y el componente de vista de textura, respectivamente. Además, el decodificador de vídeo 28 recibe el elemento sintáctico en una extensión de cabecera de fragmento.

En algunos ejemplos, el elemento sintáctico es una lista de imágenes de referencia. La lista de imágenes de referencia se puede predecir a partir de la cabecera de fragmento de referencia con un indicador de herencia de lista de imágenes de referencia, en el que el indicador de herencia de lista de imágenes de referencia con un valor de 1 indica que los elementos sintácticos de construcción de la lista de imágenes de referencia para la vista actual se copian de la cabecera del fragmento de referencia, y en el que el indicador de herencia de lista de imágenes de referencia con un valor de 0 indica que los elementos sintácticos de construcción de la lista de imágenes de referencia para la vista actual no se copian de la cabecera del fragmento de referencia. En otro ejemplo, la lista de imágenes de referencia se puede predecir a partir de la cabecera del fragmento de referencia con un indicador de inserción de lista de imágenes de referencia, en el que el indicador de inserción de lista de imágenes de referencia con un valor de 1 indica que las imágenes de referencia de la cabecera del fragmento de referencia se añaden a la referencia lista de imágenes de referencia de la vista actual. En otro ejemplo, la lista de imágenes de referencia se puede predecir a partir de la cabecera del fragmento de referencia con un indicador de exclusión de imágenes de referencia, en el que el indicador de exclusión de imágenes de referencia con un valor de 1 indica que las imágenes

de referencia de la cabecera del fragmento de referencia se excluyen de la lista de imágenes de referencia de la vista actual. En algún ejemplo, el procedimiento incluye además modificar una modificación de lista de imágenes de referencia cuando se habilita la predicción de la modificación de lista de imágenes de referencia, basándose en al menos una tabla de sintaxis de exclusión o inserción de lista de imágenes de referencia.

5 Un codificador de vídeo recibe un fragmento de textura que comprende una cabecera de fragmento de textura que comprende elementos sintácticos representativos de características del fragmento de textura (102). Por ejemplo, un codificador de vídeo recibe un fragmento de textura para un componente de vista de textura asociado con uno o más bloques codificados de datos de vídeo representativos de información de textura, comprendiendo el fragmento de  
10 textura el uno o más bloques codificados y una cabecera de fragmento de textura que comprende elementos sintácticos representativos de características del fragmento de textura. El procedimiento incluye además recibir un fragmento de profundidad que comprende una cabecera de fragmento de profundidad que comprende elementos sintácticos representativos de las características del fragmento de profundidad (104). Por ejemplo, el codificador de vídeo recibe un fragmento de profundidad para un componente de vista de profundidad asociado con uno o más bloques codificados de información de profundidad correspondientes al componente de vista de textura, en el que el  
15 fragmento de profundidad comprende el uno o más bloques codificados de información de profundidad y una cabecera de fragmento de profundidad que comprende elementos sintácticos representativos de características del fragmento de profundidad. En algunos ejemplos, tanto el componente de vista de profundidad como el componente de vista de textura pertenecen a una vista y a una unidad de acceso.

20 El procedimiento comprende además codificar un primer fragmento, en el que el primer fragmento comprende uno de entre el fragmento de textura y el fragmento de profundidad, en el que el primer fragmento tiene una cabecera de fragmento que comprende elementos sintácticos representativos de características del primer fragmento (106). Por ejemplo, el codificador de vídeo 22 codifica un primer fragmento, en el que el primer fragmento comprende uno de  
25 entre el fragmento de textura y el fragmento de profundidad, en el que el primer fragmento tiene una cabecera de fragmento que comprende elementos sintácticos representativos de características del primer fragmento. En un ejemplo, la cabecera de fragmento comprende todos los elementos sintácticos utilizados para codificar el fragmento asociado. En otro ejemplo, el decodificador de vídeo 28 decodifica un primer fragmento, en el que el primer fragmento comprende uno de entre el fragmento de textura y el fragmento de profundidad, en el que el primer  
30 fragmento tiene una cabecera de fragmento que comprende elementos sintácticos representativos de características del primer fragmento.

El procedimiento comprende además determinar elementos sintácticos comunes para un segundo fragmento a partir de la cabecera de fragmento del primer fragmento (108). Además, el procedimiento comprende codificar el segundo  
35 fragmento después de codificar el primer fragmento al menos parcialmente basándose en los elementos sintácticos comunes determinados, en el que el segundo fragmento tiene una cabecera de fragmento que comprende elementos sintácticos representativos de características del segundo fragmento que excluyen valores para elementos sintácticos que son comunes al primer fragmento (110). Por ejemplo, el codificador de vídeo 22 puede codificar el segundo fragmento después de codificar el primer fragmento al menos parcialmente basándose en los  
40 elementos sintácticos comunes determinados, en el que el segundo fragmento comprende uno de entre el fragmento de textura y el fragmento de profundidad que no es el primer fragmento, en el que el segundo fragmento tiene una cabecera de fragmento que comprende elementos sintácticos representativos de características del segundo fragmento, que excluyen valores para elementos sintácticos que son comunes al primer fragmento. De manera similar, el decodificador de vídeo 28 puede decodificar el segundo fragmento después de codificar el primer  
45 fragmento al menos parcialmente basándose en los elementos sintácticos comunes determinados, en el que el segundo fragmento comprende uno de entre el fragmento de textura y el fragmento de profundidad que no es el primer fragmento, en el que el segundo fragmento tiene una cabecera de fragmento que comprende elementos sintácticos representativos de características del segundo fragmento, que excluyen valores para elementos sintácticos que son comunes al primer fragmento.  
50

En otros ejemplos, el procedimiento comprende además señalar una indicación de qué elementos sintácticos se señalan explícitamente en la cabecera de fragmento del segundo fragmento en el conjunto de parámetros de secuencia.

55 En otros ejemplos, se determina y señala al menos un elemento sintáctico de profundidad en una cabecera de fragmento del componente de vista de profundidad. El al menos un elemento sintáctico de profundidad puede incluir un identificador de conjunto de parámetros de imagen, una diferencia de parámetro de cuantificación entre un parámetro de cuantificación del fragmento y un parámetro de cuantificación señalado en un conjunto de parámetros de imagen, una posición de inicio de la unidad de bloque codificado, un orden de las imágenes de referencia o un  
60 orden de visualización de la imagen actual del componente de vista de profundidad. Por ejemplo, la cabecera de fragmento del segundo fragmento comprende al menos un elemento sintáctico señalado de una identificación de un conjunto de parámetros de imagen de referencia. En otro ejemplo, la cabecera de fragmento del segundo fragmento comprende al menos un elemento sintáctico señalado de una diferencia de parámetro de cuantificación entre un parámetro de cuantificación del segundo fragmento y un parámetro de cuantificación señalado en un conjunto de  
65 parámetros de imagen. En otro ejemplo, la cabecera de fragmento del segundo fragmento comprende al menos un elemento sintáctico señalado de una posición de inicio del bloque codificado. Además, la cabecera de fragmento del



segundo fragmento puede comprender al menos uno de entre un número de trama y un recuento de orden de imágenes del segundo fragmento. En otro ejemplo, la cabecera de fragmento del segundo fragmento comprende al menos uno de entre los elementos sintácticos relacionados con una construcción de lista de imágenes de referencia, un número de tramas de referencia activas para cada lista, tablas de sintaxis de modificación de lista de imágenes de referencia y una tabla de ponderación de predicción .

Se puede determinar que una posición de inicio de la unidad de bloque codificado es cero cuando una posición de inicio del bloque codificado no se señala en la cabecera de fragmento de textura o en la cabecera de fragmento de profundidad. Se puede señalar un parámetro de filtro de bucle para el al menos un componente de vista de textura, y un conjunto de indicadores que indica un parámetro de filtro de bucle utilizado para el componente de vista de profundidad es igual a un parámetro de filtro de bucle para el al menos un componente de vista de textura. Por ejemplo, la cabecera de fragmento del segundo fragmento comprende al menos uno de los elementos sintácticos relacionados con unos parámetros de filtrado de eliminación de bloques o unos parámetros de filtrado de bucle adaptativo para el segundo fragmento.

En otro ejemplo, el uno o más bloques de datos de vídeo representativos de información de textura se codifican mediante predicción inter de vistas, mientras que los valores de profundidad para una parte correspondiente de la trama se codifican mediante predicción intra de vistas. Una trama de vídeo que tiene componentes de vista de textura y componentes de vista de profundidad puede corresponder a una primera vista. La codificación de uno o más bloques de datos de vídeo representativos de información de textura puede incluir predecir al menos una parte de al menos uno de los bloques de datos de vídeo representativos de información de textura en relación con unos datos de una segunda vista, en la que la segunda vista es diferente de la primera vista. La codificación de información de profundidad representativa de valores de profundidad para la parte de la trama comprende además predecir al menos una parte de la información de profundidad representativa de unos valores de profundidad en relación con los datos de la primera vista. La cabecera de fragmento de profundidad puede señalar además elementos sintácticos representativos de una construcción de lista de imágenes de referencia para el componente de vista de mapa de profundidad.

En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, firmware o cualquier combinación de estos. Si se implementan en software, las funciones pueden almacenarse en un medio legible por ordenador, o transmitirse a este, 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 corresponden a un medio tangible tal como unos medios de almacenamiento de datos, o medios de comunicación que incluyen cualquier medio que facilite la transferencia de un programa informático de un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador pueden corresponder, en general, a (1) unos medios de almacenamiento tangibles y legibles por ordenador que sean no transitorios o (2) un medio de comunicación tal como una señal o onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que se puede acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para implementar las técnicas descritas en la presente divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

A modo de ejemplo, y no de manera limitativa, dichos medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento de disco óptico, almacenamiento de disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que pueda utilizarse 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 puede denominarse de forma correcta con el término medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, un servidor u otra fuente remota mediante 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 dirigen a medios de almacenamiento tangibles no transitorios. El término disco, como se utiliza 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, de los cuales el disco flexible normalmente reproduce datos de magnéticamente, mientras que el resto de discos reproducen datos ópticamente con láseres. Las combinaciones de los anteriores también deberían incluirse dentro del alcance de los medios legibles por ordenador.

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), matrices lógicas programables in situ (FPGA) u otros circuitos lógicos integrados o discretos equivalentes. Por consiguiente, el término «procesador», como se utiliza en el presente documento, puede referirse a cualquier estructura anterior o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, las funciones descritas en el presente documento pueden estar

incluidas en módulos de hardware y/o software dedicados configurados para codificar y decodificar, o estar incorporadas en un códec combinado. Además, las técnicas podrían implementarse completamente en uno o más circuitos o elementos lógicos.

- 5 Las técnicas de la presente divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, que incluyen un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En la presente divulgación se describen varios componentes, módulos o unidades para enfatizar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no requieren necesariamente su realización mediante diferentes unidades de hardware. En su lugar, como se ha descrito anteriormente, diversas
- 10 unidades pueden combinarse en una unidad de hardware de códec o proporcionarse por medio de un grupo de unidades de hardware interoperativas, que incluyen uno o más procesadores como los descritos anteriormente, conjuntamente con software y/o firmware adecuado.

- 15 Se han descrito diversos ejemplos de la divulgación. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.

**REIVINDICACIONES**

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

5 codificar (102) uno o más bloques de datos de vídeo representativos de información de textura de al menos una parte de una trama de los datos de vídeo;

10 procesar (104) un fragmento de textura para un componente de vista de textura de una vista actual asociada con una unidad de acceso, comprendiendo el fragmento de textura el uno o más bloques codificados y una cabecera de fragmento de textura que comprende un conjunto de elementos sintácticos representativos de unas características del fragmento de textura, en el que procesar el fragmento de textura comprende al menos una de entre las etapas de generar o recibir el fragmento de textura;

15 codificar información de profundidad de codificación representativa de unos valores de profundidad para al menos la parte de la trama (106); y

20 procesar (108) un fragmento de profundidad para un componente de vista de profundidad de la vista actual correspondiente al componente de vista de textura de la vista actual, comprendiendo el fragmento de profundidad la información de profundidad codificada y una cabecera de fragmento de profundidad que comprende un conjunto de elementos sintácticos representativos de unas características del fragmento de profundidad, en el que procesar el fragmento de profundidad comprende al menos una de entre las etapas de generar o recibir el fragmento de profundidad;

25 en el que al menos una de entre:

procesar el fragmento de textura comprende:

30 codificar, en una extensión de cabecera de fragmento de la cabecera de fragmento de textura, un elemento sintáctico que indica que una cabecera de fragmento de referencia para el fragmento de textura es una cabecera de fragmento asociada con otra vista de la misma unidad de acceso que la vista actual; y

35 predecir al menos un elemento sintáctico del conjunto de elementos sintácticos de la cabecera de fragmento de textura representativos de unas características del fragmento de textura a partir de la cabecera de fragmento de referencia para el fragmento de textura, o procesar el fragmento de profundidad comprende:

40 codificar, en una extensión de cabecera de fragmento de la cabecera de fragmento de profundidad, un elemento sintáctico que indica que una cabecera de fragmento de referencia para el fragmento de profundidad es una cabecera de fragmento asociada con la otra vista u otra vista de la misma unidad de acceso que la vista actual; y procesar el fragmento de profundidad comprende predecir al menos un elemento sintáctico del conjunto de elementos sintácticos de la cabecera de fragmento de profundidad representativos de unas características del fragmento de profundidad a partir de la cabecera de fragmento de referencia.

45 2. El procedimiento según la reivindicación 1, en el que la cabecera de fragmento de referencia comprende una cabecera de fragmento de uno de entre un componente de vista de textura o un componente de vista de profundidad asociado con la otra vista.

50 3. El procedimiento según las reivindicaciones 1 o 2, en el que la cabecera de fragmento de referencia comprende una cabecera de fragmento asociada con una vista de textura básica compatible con codificación de vídeo avanzada, AVC, de la misma unidad de acceso.

55 4. El procedimiento según cualquiera de las reivindicaciones 1-3, que comprende además:

disponer el conjunto de elementos sintácticos representativos de unas características del fragmento de textura y el conjunto de elementos sintácticos representativos de unas características del fragmento de profundidad en un orden diferente de un orden de cabecera de fragmento de AVC.

60 5. El procedimiento según cualquiera de las reivindicaciones 1-4, en el que codificar el uno o más bloques de datos de vídeo y la información de profundidad comprende decodificar el uno o más bloques de datos de vídeo y la información de profundidad, en el que procesar el fragmento de textura comprende recibir el fragmento de textura, y en el que procesar el fragmento de profundidad comprende recibir el fragmento de profundidad, comprendiendo además el procedimiento:

65 recibir un índice que indica la cabecera de fragmento de referencia.

6. El procedimiento según la reivindicación 5, en el que el índice comprende un orden de vista delta, y en el que el orden de vista delta indica la ubicación de la cabecera de fragmento de referencia en relación con la vista actual.
- 5
7. El procedimiento según la reivindicación 5, que comprende además una o más de:
- 10
- recibir un indicador que indica si la cabecera de fragmento de referencia corresponde a un componente de vista de profundidad o un componente de vista de textura asociado con la otra vista;
- 15
- recibir un ID de fragmento del fragmento de profundidad o el fragmento de textura del componente de vista de profundidad o textura indicado, respectivamente;
- 20
- recibir el ID de fragmento para el fragmento de profundidad y el fragmento de textura, en el que el ID de fragmento se señala en al menos una de entre la cabecera de fragmento o una cabecera de unidad de capa de abstracción de red, NAL de una unidad NAL del componente de vista; y
- recibir el ID de fragmento para el fragmento de profundidad y el fragmento de textura, en el que el ID de fragmento se señala en una unidad de acceso que incluye una unidad NAL del componente de vista de profundidad y el componente de vista de textura, respectivamente.
8. El procedimiento según cualquiera de las reivindicaciones 1-7, que comprende además recibir el elemento sintáctico en una extensión de cabecera de fragmento.
- 25
9. El procedimiento según cualquiera de las reivindicaciones 1-8, en el que el elemento sintáctico es una lista de imágenes de referencia, en el que la lista de imágenes de referencia se predice a partir de la cabecera de fragmento de referencia con un indicador de herencia de lista de imágenes de referencia, en el que el indicador de herencia de lista de imágenes de referencia con un valor de 1 indica que unos elementos sintácticos de construcción de lista de imágenes de referencia para la vista actual se copian de la cabecera de fragmento de referencia, y en el que el indicador de herencia de lista de imágenes de referencia con un valor de 0 indica que unos elementos sintácticos de construcción de lista de imágenes de referencia para la vista actual no se copian de la cabecera de fragmento de referencia.
- 30
10. El procedimiento según la reivindicación 1, en el que el elemento sintáctico es una lista de imágenes de referencia, en el que la lista de imágenes de referencia se predice a partir de la cabecera de fragmento de referencia con un indicador de inserción de lista de imágenes de referencia, en el que el indicador de inserción de lista de imágenes de referencia con un valor de 1 indica que unas imágenes de referencia de la cabecera de fragmento de referencia se añaden a la lista de imágenes de referencia de la vista actual.
- 35
11. El procedimiento según la reivindicación 1, en el que el elemento sintáctico es una lista de imágenes de referencia, en el que la lista de imágenes de referencia se predice a partir de la cabecera de fragmento de referencia con un indicador de exclusión de imágenes de referencia, en el que el indicador de exclusión de imágenes de referencia con un valor de 1 indica que unas imágenes de referencia de la cabecera de fragmento de referencia se excluyen de la lista de imágenes de referencia de la vista actual.
- 40
- 45
12. El procedimiento según cualquiera de las reivindicaciones 1-8, que comprende además:
- 50
- modificar una modificación de lista de imágenes de referencia cuando la predicción de la modificación de lista de imágenes de referencia está habilitada, basándose en al menos una de entre una tabla de sintaxis de exclusión o inserción de lista de imágenes de referencia.
- 55
13. El procedimiento según la reivindicación 1, en el que codificar el uno o más bloques de datos de vídeo y la información de profundidad comprende codificar el uno o más bloques de datos de vídeo y la información de profundidad, en el que procesar el fragmento de textura comprende generar el fragmento de textura, y en el que procesar el fragmento de profundidad comprende generar el fragmento de profundidad, comprendiendo además el procedimiento:
- 60
- generar un índice que indica la cabecera de fragmento de referencia, en el que el índice comprende un orden de vista delta, y en el que el orden de vista delta indica la ubicación de la cabecera de fragmento de referencia en relación con la vista actual;
- 65
- generar un indicador que indica si la cabecera de fragmento de referencia corresponde a un componente de vista de profundidad o un componente de vista de textura asociado con la otra vista; y
- generar un ID de fragmento del fragmento de profundidad o el fragmento de textura del componente de vista de profundidad o de textura indicado, respectivamente, en el que el ID de fragmento se genera en al

menos una de entre la cabecera de fragmento, una cabecera de unidad de capa de abstracción de red, NAL, de una unidad NAL del componente de vista, o una unidad de acceso que incluye una unidad de capa de abstracción de red, NAL, del componente de vista de profundidad y el componente de vista de textura, respectivamente.

- 5
- 14.** Un medio de almacenamiento legible por ordenador que tiene instrucciones almacenadas que, al ejecutarse, hacen que un procesador de un dispositivo de codificación de vídeo realice el procedimiento de cualquiera de las reivindicaciones 1-13.
- 10
- 15.** Un dispositivo para procesar datos de vídeo, que comprende medios para realizar el procedimiento de cualquiera de las reivindicaciones 1-13.

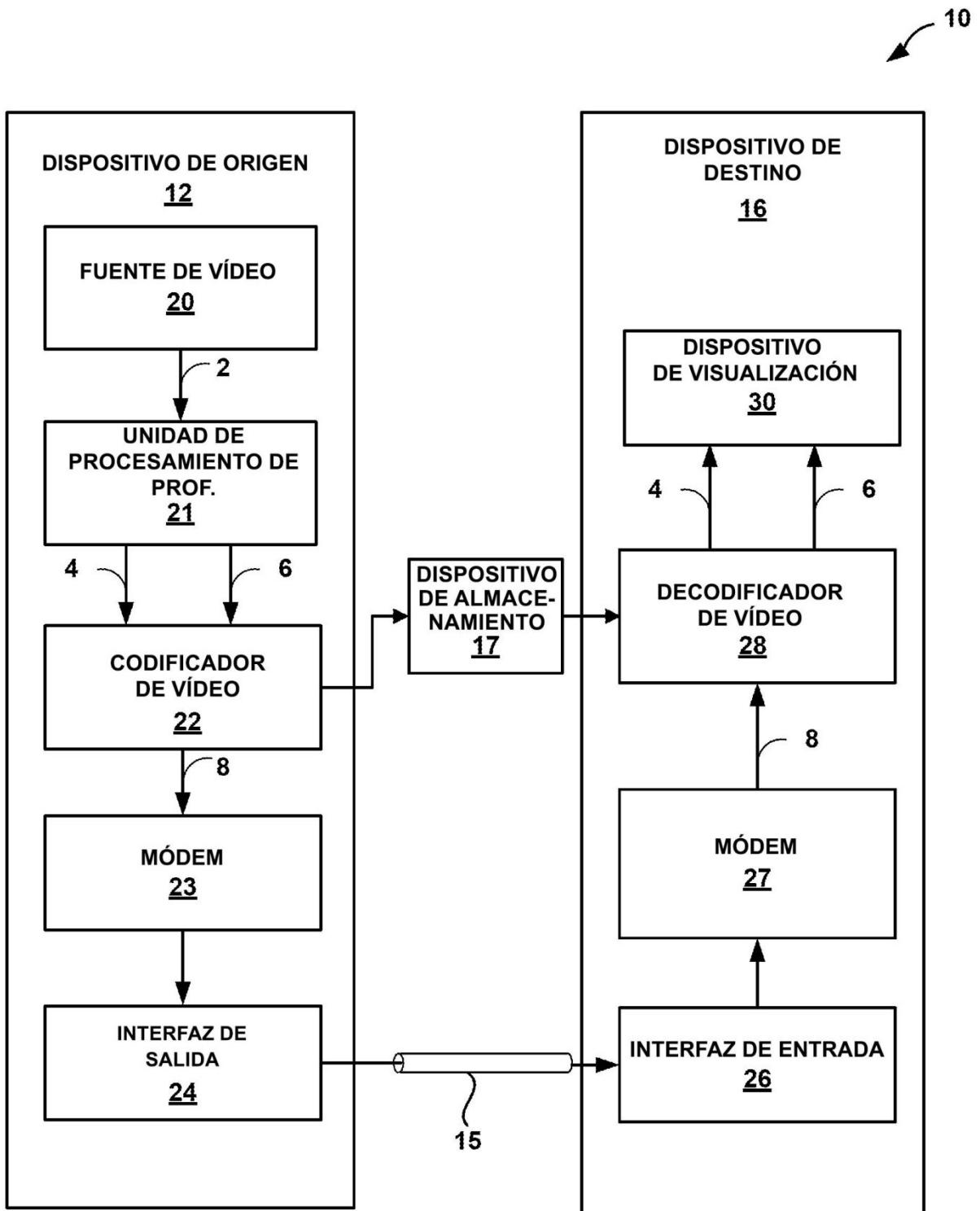


FIG. 1

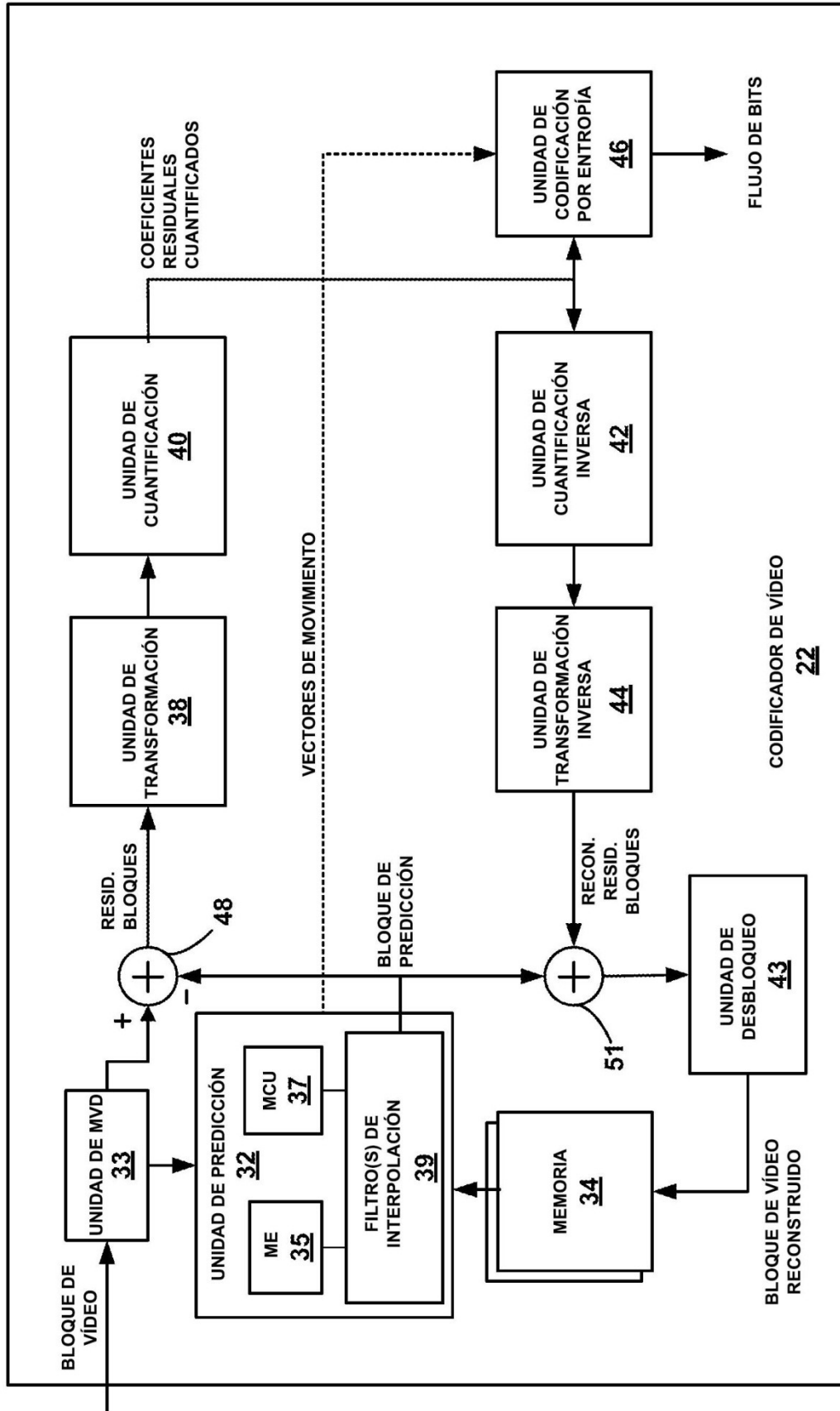
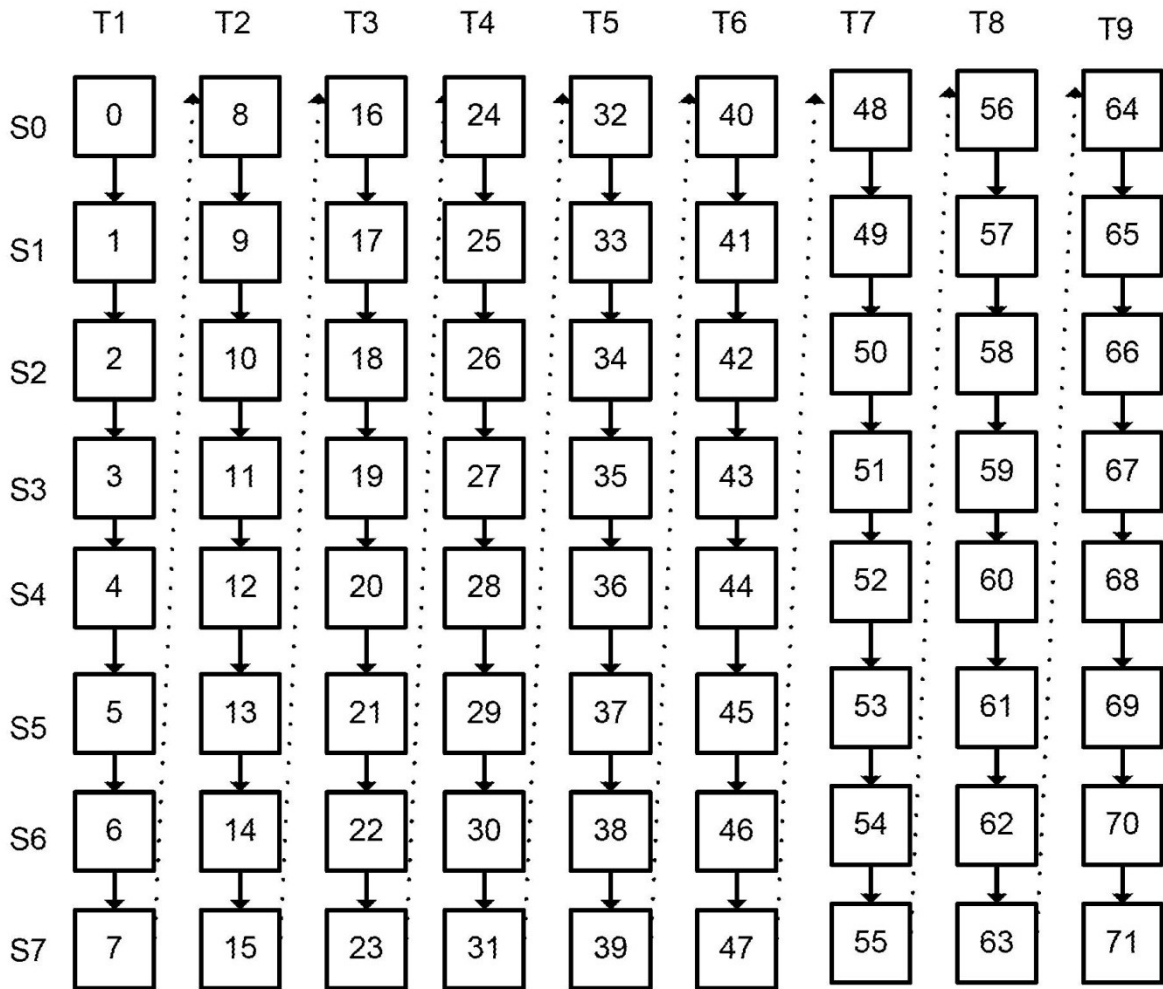


FIG. 2



**FIG. 3**



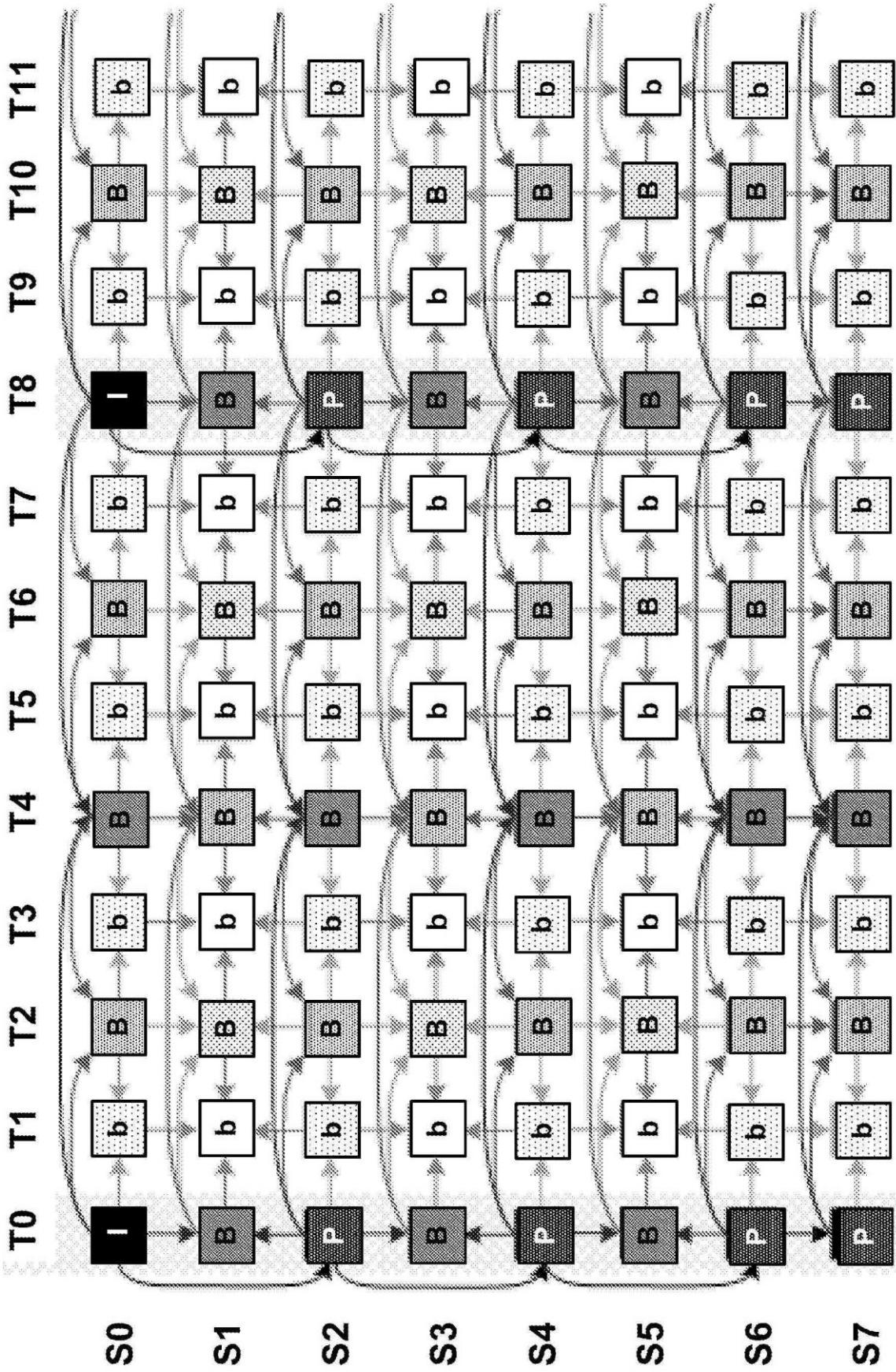


FIG. 4

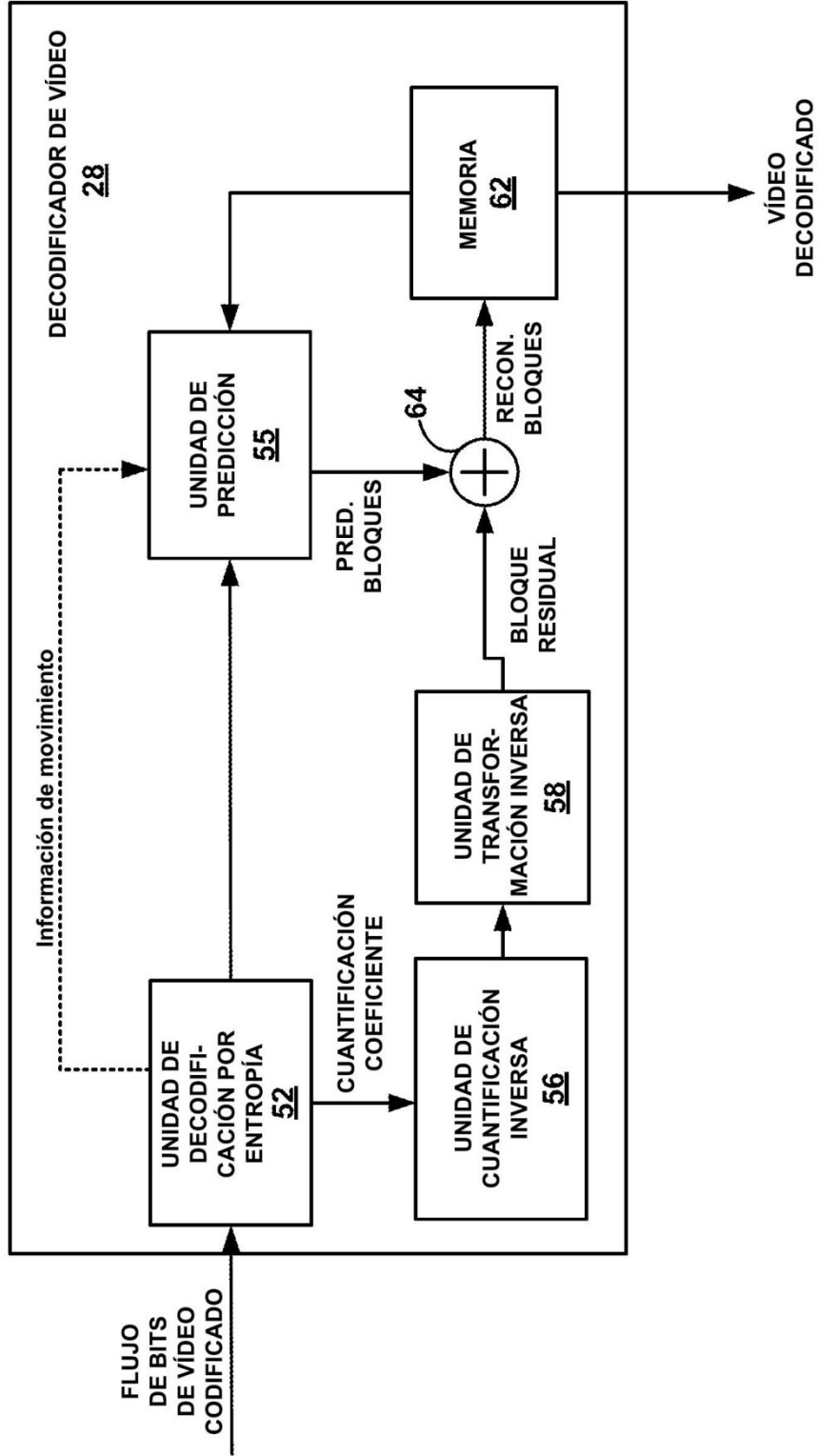


FIG. 5

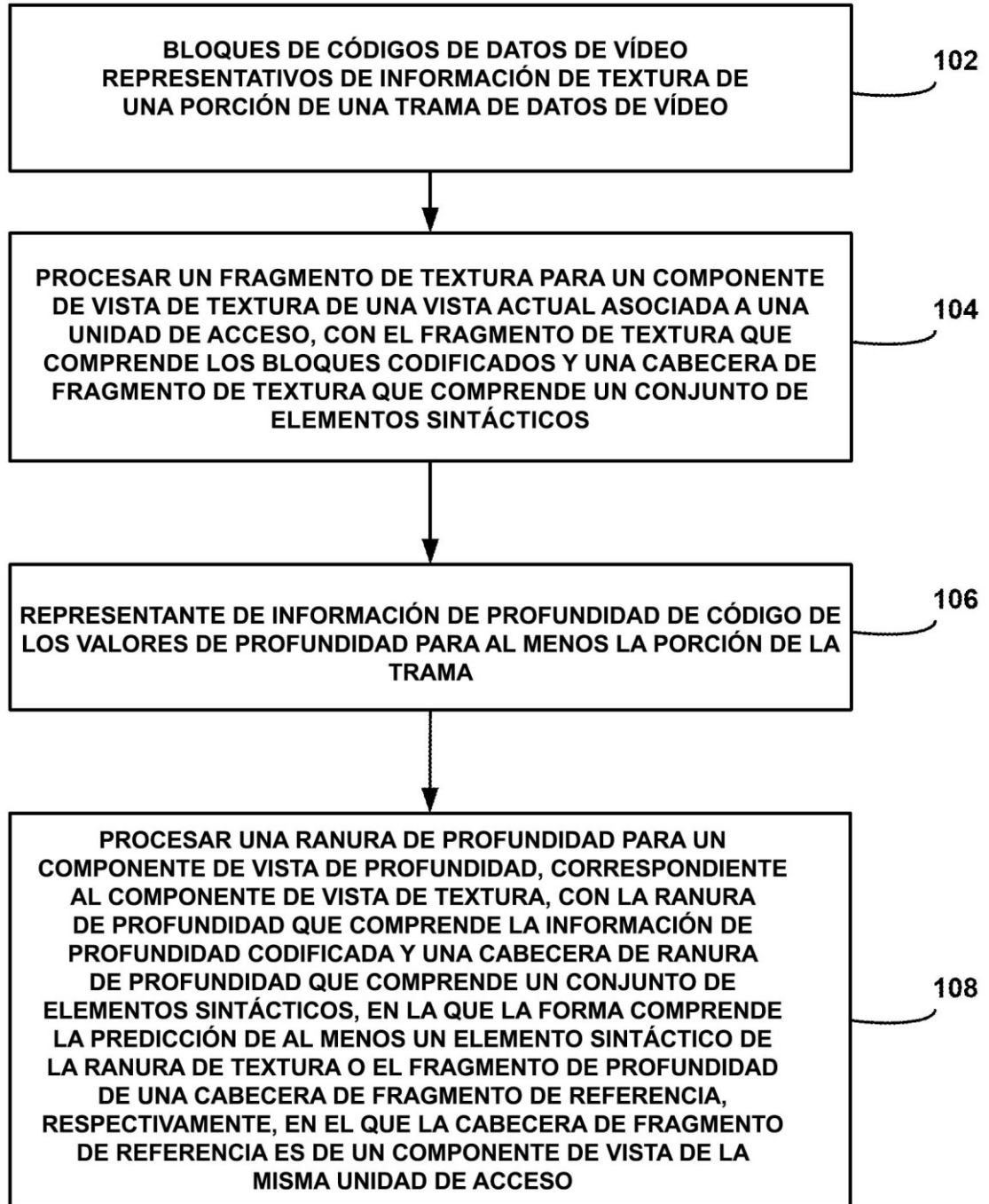


FIG. 6