

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 692 721**

51 Int. Cl.:

H04N 13/00 (2008.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **08.03.2012 PCT/US2012/028252**

87 Fecha y número de publicación internacional: **13.09.2012 WO12122364**

96 Fecha de presentación y número de la solicitud europea: **08.03.2012 E 12708492 (9)**

97 Fecha y número de publicación de la concesión europea: **01.08.2018 EP 2684372**

54 Título: **Codificación de vídeo de múltiples visualizaciones más contenido de profundidad**

30 Prioridad:

10.03.2011 US 201161451333 P
18.05.2011 US 201161487587 P
07.03.2012 US 201213414515

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
04.12.2018

73 Titular/es:

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

72 Inventor/es:

CHEN, YING;
ZHANG, RONG y
KARCZEWICZ, MARTA

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 692 721 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

Codificación de vídeo de múltiples visualizaciones más contenido de profundidad

5 **CAMPO TÉCNICO**

[0001] Esta divulgación se refiere al campo de la codificación de vídeo, específicamente, la codificación de contenido de vídeo tridimensional.

10 **ANTECEDENTES**

[0002] 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.

[0003] 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 en 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 en la misma trama o fragmento, o predicción temporal con respecto a otras tramas de referencia.

[0004] 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.

[0005] 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 ajustable a escala (SVC), que es la extensión ajustable a escala para la H.264/AVC. Otra norma es la codificación de vídeo de múltiples visualizaciones (MVC), que se ha convertido en la extensión de múltiples visualizaciones 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" ["Borrador conjunto 8.0 sobre codificación de vídeo de múltiples visualizaciones"], 28.ª reunión de JVT, Hanover, 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 JVT-AD007, "Revisión del borrador de los editores de Rec. ITU-T. H.264 | ISO/IEC 14496-10 Codificación avanzada de vídeo: en preparación para el consentimiento de ITU-T SG 16 AAP (en forma integrada)", "30.ª reunión de JVT, 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.

45 **RESUMEN**

[0006] En general, esta divulgación describe técnicas para soportar la codificación de datos de vídeo que incluyen información de textura y profundidad. La información de textura se refiere a la información para renderizar una imagen bidimensional, como la información de luminancia (brillo) y crominancia (color). La información de profundidad puede incluir información útil para formar vídeo tridimensional, tal como un mapa de profundidad (por ejemplo, valores de profundidad por píxel) o un mapa de paralaje (por ejemplo, disparidad horizontal por píxel).

[0007] En general, la presente divulgación describe unas técnicas para soportar la renderización de vídeo tridimensional (3D). En particular, las técnicas de esta divulgación se refieren a la codificación y decodificación de contenido de vídeo 3D que tiene componentes de textura y profundidad. Esta divulgación también propone técnicas de señalización para datos de vídeo. Los componentes de profundidad pueden ser de un mapa de profundidad de una imagen. En la renderización de imágenes 3D, los mapas de profundidad incluyen componentes de profundidad y se pueden usar para generar visualizaciones virtuales desde una perspectiva de visualización proporcionada. Las técnicas de señalización pueden incluir señalización de los componentes de visualización de textura y profundidad, dependencia de predicción de componentes de visualización de profundidad, la dependencia del componente de visualización de profundidad y su componente de visualización de textura correspondiente dentro de un componente de visualización, y si un componente de visualización de textura puede depender del componente de visualización de profundidad de otra instancia de tiempo para, por ejemplo, predicción de síntesis de visualización. La señalización también puede incluir proporcionar parámetros de cámara asociados con cada visualización. Las unidades de bloques codificados, denominadas también simplemente "bloques codificados" en esta divulgación, pueden corresponder a

macrobloques en ITU-T H.264/AVC (Codificación de vídeo avanzada) o unidades de codificación de Codificación de vídeo de alta eficiencia (HEVC).

5 [0008] En todos los ejemplos de esta divulgación, la textura NAL y la profundidad NAL de un componente de visualización se organizan de manera consecutiva (por ejemplo, orden de codificación de unidad VLC NAL) en un flujo de bits común y se utilizan técnicas de señalización para indicar con la cabecera de la unidad VCL NAL si la unidad NAL pertenece a un componente de visualización de profundidad, prediciendo desde el componente de visualización de textura hasta el componente de visualización de profundidad correspondiente. En algunos ejemplos, pueden señalarse parámetros relacionados con el contenido de la cámara y la profundidad. Los parámetros pueden incluir, por ejemplo, parámetros de cámara intrínsecos comunes y parámetros de cámara extrínsecos, asignación desde el valor de profundidad cuantificado al valor de profundidad del mundo real, diferentes parámetros de cámara extrínsecos (por ejemplo, la traslación de cada visualización), etc.

15 [0009] En un ejemplo, la divulgación se dirige a un procedimiento de codificación de vídeo de acuerdo con la reivindicación 1.

[0010] En otro ejemplo, la divulgación se dirige a un dispositivo para procesar datos de vídeo de acuerdo con la reivindicación 5.

20 [0011] 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 circuitería lógica integrada o discreta 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.

[0012] En otro ejemplo, esta divulgación se dirige a un procedimiento de decodificación de datos de vídeo de acuerdo con la reivindicación 9.

30 [0013] En otro ejemplo más, esta divulgación se dirige a un dispositivo para procesar datos de vídeo de acuerdo con la reivindicación 11.

35 [0014] En otro ejemplo más, esta divulgación se dirige a un producto de programa informático de acuerdo con la reivindicación 13.

[0015] 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 esta divulgación resultarán evidentes a partir de la descripción y los dibujos, y a partir de las reivindicaciones.

40 **BREVE DESCRIPCIÓN DE LOS DIBUJOS**

[0016]

45 La FIG. 1 es un diagrama de bloques que ilustra un ejemplo de un sistema de codificación y decodificación de vídeo que puede usarse para implementar una o más de las técnicas de esta divulgación.

La FIG. 2 es un diagrama de bloques que ilustra con mayor detalle un ejemplo del codificador de vídeo de la FIG. 1.

50 La FIG. 3A es un diagrama conceptual que ilustra un ejemplo de una estructura de flujo de bits que puede usarse en una implementación de una o más de las técnicas de esta divulgación.

La FIG. 3B es un diagrama conceptual que ilustra un ejemplo de una visualización que puede incluirse en la estructura de flujo de bits de la FIG. 3A.

55 La FIG. 4A es un diagrama conceptual que ilustra un patrón de predicción de MVC de ejemplo.

60 La FIG. 4B es un diagrama conceptual que ilustra la codificación de vídeo de múltiples visualizaciones (MVC) compatible con la codificación de vídeo de múltiples visualizaciones más profundidad (MVD) para dos visualizaciones.

65 La FIG. 4C es un diagrama conceptual que ilustra la codificación MVD compatible con MVC para tres o más visualizaciones.

La FIG. 4D es un diagrama conceptual que ilustra la codificación MVD compatible con MVC para múltiples visualizaciones.

5 La FIG. 5 es un diagrama de flujo que ilustra una operación de ejemplo de un codificador de vídeo coherente con esta divulgación.

La FIG. 6 es un diagrama de bloques que ilustra un ejemplo del descodificador de vídeo de la FIG. 1 en más detalle.

10 La FIG. 7 es un diagrama de flujo que ilustra un funcionamiento de ejemplo de un descodificador de vídeo coherente con esta divulgación.

DESCRIPCIÓN DETALLADA

15 **[0017]** Esta divulgación describe técnicas para soportar la codificación de datos de vídeo que incluyen información de textura y profundidad. La información de textura se refiere a la información para renderizar una imagen bidimensional (2D), como la información de luminancia (brillo) y crominancia (color). La información de profundidad puede incluir información útil en la formación de vídeo 3D, tal como un mapa de profundidad (por ejemplo, valores de profundidad por píxel) o un mapa de paralaje (por ejemplo, disparidad horizontal por píxel).

20 **[0018]** Las técnicas descritas están relacionadas con la codificación de contenido de vídeo tridimensional (3D). En particular, estas técnicas están dirigidas a proporcionar componentes de profundidad para componentes de textura de una o más visualizaciones de un flujo de bits MVC. El contenido de vídeo 3D se puede representar, por ejemplo, como datos de vídeo más profundidad (MVD) multivisión. Es decir, estas técnicas pueden aplicarse para codificar un flujo de bits que se asemeja a un flujo de bits de codificación de vídeo de múltiples visualizaciones (MVC), donde cualquiera o todas las visualizaciones del flujo de bits MVC pueden incluir información de profundidad, por ejemplo, componentes de visualización de profundidad.

30 **[0019]** La conversión de vídeo basada en la estimación de la profundidad y la síntesis de visualización virtual se puede utilizar para crear imágenes 3D, como aplicaciones de vídeo 3D. En particular, las visualizaciones virtuales de una escena se pueden utilizar para crear una visualización 3D de la escena. La generación de una visualización virtual de una escena basada en una visualización existente de la escena se consigue convencionalmente estimando valores de profundidad de objeto antes de sintetizar la visualización virtual. La estimación de profundidad es el proceso de estimación de distancias absolutas o relativas entre objetos y el plano de cámara a partir de pares estéreo o contenido monoscópico. La información de profundidad puede incluir información útil para formar vídeo tridimensional, tal como un mapa de profundidad (por ejemplo, valores de profundidad por píxel) o un mapa de paralaje (por ejemplo, disparidad horizontal por píxel).

40 **[0020]** La información de profundidad estimada, normalmente representada por un mapa de profundidad de imagen en escala de grises, puede utilizarse para generar un ángulo arbitrario de visualizaciones virtuales mediante técnicas de renderización basadas en imagen de profundidad (DIBR). En comparación con los sistemas tradicionales de televisión tridimensional (3DTV) en los que las secuencias de múltiples visualizaciones afrontan los retos de la compresión inter de visualizaciones eficiente, un sistema basado en mapa de profundidad puede reducir el uso del ancho de banda transmitiendo solo una o varias visualizaciones junto con el (los) mapa(s) de profundidad, que pueden codificarse eficazmente. Los mapas de profundidad utilizados en la conversión basada en el mapa de profundidad pueden ser controlables (por ejemplo, a través de la escala) por los usuarios finales antes de ser utilizados en la síntesis de visualización. Pueden generarse visualizaciones 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 una visualización 2D.

50 **[0021]** MVC es una extensión de la norma de codificación ITU-T H.264 que soporta la codificación de dos o más visualizaciones de una escena. También se ha propuesto una extensión MVC de HEVC. Al capturar datos de vídeo en bruto, dos o más cámaras pueden estar activas durante la filmación de una escena, y cada una de las cámaras puede producir datos para una visualización respectiva desde una perspectiva diferente. De acuerdo con MVC, cada una de las visualizaciones puede estar codificada, y las visualizaciones pueden codificarse utilizando imágenes de otras visualizaciones codificadas como imágenes de referencia (predicción entre visualizaciones), así como otras imágenes de la misma visualización como imágenes de referencia (predicción temporal), como se ilustra en la FIG. 4A.

60 **[0022]** Las diferentes visualizaciones, capturadas por las dos o más cámaras, se pueden usar para soportar datos de vídeo 3D. Se pueden visualizar dos visualizaciones simultáneamente usando diferentes polarizaciones de luz, y un espectador puede usar gafas polarizadas pasivas de modo que cada uno de los ojos del espectador reciba una de las visualizaciones respectivas. De forma alternativa, el espectador puede llevar gafas activas que obturan cada ojo de forma independiente, y una pantalla puede alternar rápidamente entre las imágenes de cada ojo de forma sincronizada con las gafas. Algunas pantallas son capaces de mostrar hasta ocho o incluso más visualizaciones, de modo que los usuarios pueden observar la pantalla desde varios ángulos para ver la escena desde varias perspectivas.

[0023] Las técnicas implican la codificación de información de textura y profundidad en un flujo de bits común para la misma visualización. Un descodificador de vídeo puede generar una o más visualizaciones adicionales a partir de la información de textura y profundidad de una visualización codificada. Por lo tanto, en lugar de codificar un conjunto completo de ocho visualizaciones, por ejemplo, un codificador de acuerdo con las técnicas de esta divulgación puede codificar, por ejemplo, cuatro visualizaciones, que incluyen información de textura y profundidad para cada visualización. En esta divulgación, el término "codificación" puede referirse a cualquiera de los procesos de codificación y descodificación o a ambos.

[0024] Más específicamente, las técnicas de esta divulgación están dirigidas a soportar la codificación de información de textura y profundidad de la misma visualización dentro del mismo flujo de bits. Las técnicas existentes para codificar información de textura y profundidad están dirigidas a proporcionar la información de textura y profundidad en flujos de bits separados.

[0025] En un aspecto de esta divulgación, el texto y la información de profundidad pueden codificarse como componentes independientes del mismo flujo de bits, lo cual puede permitir que los dos componentes se separen fácilmente entre sí. Como resultado, cuando el flujo de bits se envía a dispositivos de visualización, el flujo de bits se puede ajustar de acuerdo con las capacidades del dispositivo de visualización. Por ejemplo, el flujo de bits que incluye ambos componentes puede enviarse a dispositivos listos para 3D, mientras que el flujo de bits puede filtrarse y reenviarse a dispositivos preparados para 2D sin incluir el componente de información de profundidad.

[0026] Más particularmente, esta divulgación propone proporcionar datos para una visualización de datos de vídeo de modo que un componente de visualización de la visualización incluye tanto información de textura como de profundidad. Un componente de visualización es una unidad de datos que incluye todos los datos para una instancia temporal común para una visualización particular. Una unidad de capa de abstracción de red (NAL) en la capa de codificación de vídeo (VCL), también denominada unidad VCL NAL, puede incluir un fragmento o trama de datos de vídeo codificados, por ejemplo, datos de textura o profundidad. Las técnicas de esta divulgación proponen tratar las unidades VCL NAL de textura y profundidad para una instancia temporal común como perteneciente a un componente de visualización común.

[0027] De acuerdo con las técnicas de esta divulgación, al proporcionar unidades VCL NAL de textura y profundidad para una instancia temporal común en el mismo componente de visualización, ciertos datos de codificación para la información de textura pueden reutilizarse para codificar la información de profundidad. Por ejemplo, un mapa de profundidad para una unidad VCL NAL puede predecirse a partir de la información de textura correspondiente usando vectores de desplazamiento. Al igual que con MVC convencional, la predicción entre visualizaciones también puede estar disponible para codificar información de textura de visualizaciones relativas a la información de textura de otras visualizaciones.

[0028] Adicionalmente, esta divulgación proporciona técnicas para señalar parámetros de cámara para visualizaciones que se proporcionan, y usar los parámetros de cámara en visualización de predicción de síntesis. Usando los parámetros de la cámara, un descodificador de vídeo puede sintetizar una visualización virtual desde una visualización de referencia y un mapa de profundidad correspondiente usando, por ejemplo, renderización basada en imágenes de profundidad (DIBR). La visualización virtual sintetizada se puede usar como una predicción para codificar otra visualización. En un ejemplo, la visualización de referencia puede ser el mapa de profundidad en sí. De esta manera, los parámetros de la cámara y un mapa de profundidad de referencia pueden ser suficientes para sintetizar un mapa de profundidad actual virtual.

[0029] Las técnicas descritas en el presente documento se pueden aplicar para codificar componentes de textura y profundidad y encapsularlos juntos en un solo flujo de bits. Los componentes de textura y profundidad asociados con una sola visualización en un solo instante de tiempo pueden codificarse juntos y asociarse con una unidad de datos de vídeo, por ejemplo, un fragmento o trama de vídeo. De acuerdo con las técnicas de esta divulgación, un codificador de vídeo puede reutilizar los datos de predicción asociados con el componente de textura de una visualización para codificar el componente de profundidad de la visualización. De esta manera, el codificador puede codificar componentes de textura y profundidad usando dependencias de predicción entre los dos componentes o dos visualizaciones, o en visualizaciones sintetizadas, y señalar las dependencias de predicción que serán útiles para codificar los datos de vídeo. Adicionalmente, esta divulgación puede proporcionar técnicas de predicción entre visualizaciones para codificar información de textura de visualizaciones relativas a la información de textura de otras visualizaciones. Estas y otras técnicas de esta divulgación se pueden describir con más detalle a continuación.

[0030] La FIG. 1 es un diagrama de bloques que ilustra un ejemplo de un sistema de codificación y descodificación de vídeo 10 que puede usarse para implementar una o más de las técnicas de esta divulgación. Como la FIG. 1 muestra, el sistema 10 incluye un dispositivo de origen 12 que transmite vídeo codificado a un dispositivo de destino 16 a través de un canal de comunicación 15. El dispositivo de origen 12 y el dispositivo de destino 16 pueden comprender cualquiera de una amplia gama de dispositivos. En algunos casos, el dispositivo de origen 12 y el dispositivo de destino 16 pueden comprender dispositivos de comunicación inalámbrica, tales como microteléfonos inalámbricos, los denominados radioteléfonos celulares o vía satélite, o cualquier dispositivo inalámbrico que pueda comunicar información de vídeo a través de un canal de comunicación 15, en cuyo caso el canal de comunicación 15

es inalámbrico en este caso. Sin embargo, las técnicas de la presente divulgación, que se refieren a la codificación 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. Por ejemplo, el vídeo codificado puede almacenarse en un dispositivo de almacenamiento para su posterior recuperación, descodificación y visualización mediante el dispositivo de destino 16. Además, las técnicas de codificación o descodificación también pueden aplicarse en un dispositivo autónomo que no necesariamente se comunica con otro dispositivo.

[0031] En el ejemplo de la FIG. 1, el dispositivo de origen 12 puede incluir una fuente de vídeo 20, una unidad de procesamiento de profundidad 21, un codificador de vídeo 22, un modulador/desmodulador (módem) 23 y un transmisor 24. El dispositivo de destino 16 puede incluir un receptor 26, un módem 27, un descodificador 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 descodificador 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 descodificación de vídeo.

[0032] 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 ondulada, transformadas enteras u otros tipos de transformadas. A título de ejemplo, en un proceso de DCT, un conjunto de valores de píxel puede convertirse en coeficientes de transformada, que pueden representar 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 por 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 modos de codificación, información de movimiento, patrones de bloques codificados y coeficientes de transformada cuantificados. Entre los ejemplos de codificación por 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).

[0033] 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. Usando los vectores de movimiento, el descodificador de vídeo 28 puede reconstruir los bloques predictivos que se usaron 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 descodificador 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 codifica la predicción inter), indicaciones de modos de predicción intra para la predicción intra, información sobre componentes de profundidad y textura, y posiblemente otros datos, tales como tipos de elementos sintácticos.

[0034] 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 puede predecir espacialmente los bloques. El codificador de vídeo 22 puede configurarse con una variedad de modos 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 un modo de predicción intra basándose en un componente de luminancia de un bloque y reutilizar, a continuación, el modo de predicción intra para codificar componentes de crominancia del bloque. Además, de acuerdo con las técnicas de esta divulgación, el codificador de vídeo 22 puede reutilizar la relación de predicción utilizada para codificar un componente de textura, que comprende componentes de luma y croma, de una visualización para codificar un componente de profundidad de la visualización. Adicionalmente, de acuerdo con las técnicas de esta divulgación, el codificador de vídeo 22 puede codificar un componente de textura de una visualización con relación a componentes de textura de otras visualizaciones, y puede codificar un componente de profundidad de una visualización con relación a componentes de profundidad de otras visualizaciones. En algunos ejemplos, el codificador de vídeo 22 puede codificar el componente de profundidad con relación a los datos de textura de una visualización sintetizada.

[0035] Las técnicas de esta divulgación pueden simplificar el proceso de codificación de componentes de textura y profundidad en MVD, y pueden mejorar la eficiencia del flujo de bits encapsulando componentes de textura y profundidad asociados con una visualización en el mismo flujo de bits. Estas técnicas también pueden incluir el suministro de información para la predicción intra-visualización, la predicción entre visualizaciones y la predicción de síntesis de visualización. En la predicción intra-visualización, se puede señalar información sobre la predicción del

movimiento desde textura hasta profundidad. En la predicción entre visualizaciones, se puede señalar información con respecto a la predicción de una visualización a la otra. La predicción de síntesis de visualización puede permitir la síntesis de una visualización virtual desde una visualización de referencia que incluya componentes de textura y profundidad.

5 **[0036]** El sistema ilustrado 10 de la FIG. 1 es solo un ejemplo. Las diversas técnicas de esta divulgación se pueden realizar mediante cualquier dispositivo de codificación que soporte codificación predictiva, o mediante cualquier dispositivo de descodificación que soporte descodificación predictiva. 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 10
10 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 descodificación de vídeo. Por lo tanto, el sistema 10 puede soportar 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.

15 **[0037]** La fuente de vídeo 20 del dispositivo de origen 12 puede incluir un dispositivo de captura de vídeo, tal como una cámara de vídeo, un archivo de vídeo que contenga vídeo capturado previamente o vídeo procedente de un proveedor de contenido de vídeo. De forma alternativa adicional, la fuente de vídeo 20 puede generar datos, basándose en gráficos por ordenador, como el 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 capturado, pre-capturado o generado por ordenador puede codificarse mediante el codificador de vídeo 22. La fuente de vídeo 20 captura una visualización 2 y la proporciona a la unidad de procesamiento de profundidad 21.
20
25

[0038] La fuente de vídeo 20 proporciona la visualización 2 a la unidad de procesamiento de profundidad 21 para calcular la imagen de profundidad de los objetos de la visualización 2. La unidad de procesamiento de profundidad 21 puede configurarse para calcular automáticamente valores de profundidad para objetos en la imagen de la visualización 2. Por ejemplo, la unidad de procesamiento de profundidad 21 puede calcular valores de profundidad para objetos basándose en información de textura, que puede incluir información de luminancia y crominancia. En algunos ejemplos, la unidad de procesamiento de profundidad 21 puede configurarse para recibir información de profundidad de un usuario. En algunos ejemplos, la fuente de vídeo 20 puede capturar dos o más visualizaciones de una escena en diferentes perspectivas, y luego calcular la información de profundidad para los objetos en la escena basándose en la disparidad entre los objetos en las dos o más visualizaciones. En varios ejemplos, la fuente de vídeo 20 puede comprender una cámara bidimensional estándar, un sistema de dos o más cámaras que proporciona una visualización estereoscópica de una escena, un conjunto de cámaras que capta varias visualizaciones de la escena o una cámara que capta una visualización más información de profundidad.
30
35

[0039] La unidad de procesamiento de profundidad 21 puede pasar los componentes de textura 4 y la información de profundidad 6 al codificador de vídeo 22. La unidad de procesamiento de profundidad 21 también puede pasar la visualización 2 al codificador de vídeo 22. La información de profundidad 6 puede comprender una imagen de mapa de profundidad para la visualización 2. Un 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 región de píxeles puede ser 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, puede haber múltiples componentes de profundidad por píxel.
40
45

[0040] En algunos ejemplos, el mapa de profundidad es estimado. Cuando hay más de una visualización presente, la coincidencia estéreo se puede usar para estimar los mapas de profundidad cuando hay más de una visualización disponible. 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 la renderización 3D basado en el renderización basado en imagen de profundidad (DIBR).
50

[0041] Aunque la fuente de vídeo 20 puede proporcionar múltiples visualizaciones de escena, la unidad de procesamiento de profundidad 21 puede calcular información de profundidad basándose en las múltiples visualizaciones y el dispositivo de origen 12 puede transmitir en general una visualización más información de profundidad para cada visualización de una escena. En un ejemplo de esta divulgación, los parámetros asociados con las diferentes visualizaciones, por ejemplo, los parámetros de la cámara pueden ser transmitidos y usados posteriormente por el descodificador para sintetizar una visualización virtual utilizando una visualización de referencia y un mapa de profundidad correspondiente. Los parámetros de la cámara pueden incluir, por ejemplo, parámetros intrínsecos (por ejemplo, la distancia focal y el desplazamiento del punto principal en la dirección horizontal) y parámetros extrínsecos (por ejemplo, una ubicación de cámara horizontal del mundo real para cada visualización). Típicamente, cada visualización de una secuencia de múltiples visualizaciones compartirá los mismos parámetros intrínsecos. Por lo tanto, estos parámetros pueden señalarse una vez para la secuencia, de modo que los parámetros intrínsecos señalados se apliquen a todas las visualizaciones de la secuencia. De esta manera, para cada secuencia,
55
60
65

ciertos parámetros de la cámara pueden señalarse y comunicarse una vez, reduciendo así la cantidad de información codificada y transmitida. En un ejemplo, los parámetros de la cámara pueden señalarse a nivel de secuencia.

[0042] En un ejemplo, cada imagen de mapa de profundidad puede representarse utilizando píxeles de 8 bits. Para cada píxel de 8 bits de la imagen de mapa de profundidad, se puede usar una asignación de valores z del mundo real al valor de profundidad de 8 bits cuantificado, por ejemplo, para DIBR, que puede ser parte del circuito de descodificación. En el nivel de secuencia o nivel de imagen, la función que se asigna desde el valor de profundidad de 8 bits hasta el valor de z del mundo real puede señalarse o asumirse. En este ejemplo, las funciones por defecto de conversión de valor de profundidad a z pueden señalarse; por ejemplo, una indicación para especificar que la función es una función proporcional inversa desde el valor de profundidad hasta el valor z. Además, cuando se decide una función, el rango de profundidad también se puede señalar, por ejemplo, en el conjunto de parámetros de secuencia. En los ejemplos donde se utiliza una conversión más compleja entre el valor de profundidad y el valor z, se puede señalar una tabla que contiene el valor de profundidad al valor z del mundo real (256 números flotantes).

[0043] Cuando la visualización 2 es una imagen digital fija, el codificador de vídeo 22 puede configurarse para codificar la visualización 2 como, por ejemplo, una imagen del Grupo conjunto de expertos en fotografía (JPEG). Cuando la visualización 2 es una trama de datos de vídeo, el codificador de vídeo 22 puede estar configurado para codificar la primera visualización 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 (IEC) MPEG-1 Visual, la ISO/IEC MPEG-4 Visual, la Unión internacional de telecomunicaciones (ITU) H.261, ITU-T H.262, ITU-T H.263, ITU-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 codificar, para cada visualización, datos de textura y datos de profundidad (por ejemplo, fragmentos de textura y fragmentos de profundidad). El codificador de vídeo 22 puede formar unidades de capa de abstracción de red (NAL) para los fragmentos de textura y los fragmentos de profundidad y encapsular las unidades NAL de una visualización particular en una instancia temporal particular como un componente de visualización. De esta manera, el flujo de bits codificada 8 puede incluir datos de imágenes codificadas o información de textura 4, junto con la información de profundidad 6. El codificador de vídeo 22 pasa el flujo de bits codificado 8 al transmisor 24. El flujo de bits codificado 8 puede incluir información de textura codificada 4, información de profundidad 6, y puede transmitirse junto con información señalada tal como, por ejemplo, dependencias de predicción y parámetros de cámara.

[0044] El flujo de bits codificado 8 puede incluir información de textura 4 e información de profundidad 6. Los componentes de textura 4 pueden incluir componentes de luminancia (luma) y cromaticidad (croma) de la información de vídeo. Los componentes de luma en general describen el brillo, mientras que los componentes de cromaticidad en general describen tonalidades de color. La unidad de procesamiento de profundidad 21 puede extraer información de profundidad 6 de un mapa de profundidad del vídeo 2. El codificador de vídeo 22 puede codificar los componentes de textura 4 y la información de profundidad 6 y encapsularlos en un solo flujo de bits, en particular, el flujo de bits codificado 8. Asimismo, el codificador de vídeo 22 puede codificar el flujo de bits de manera que las dependencias de predicción de movimiento para el componente de textura se usan para codificar el componente de profundidad. De esta manera, el componente de profundidad puede predecirse a partir del componente de textura. Adicionalmente, de acuerdo con las técnicas de esta divulgación, el codificador de vídeo 22 puede señalar parámetros de cámara asociados con una visualización particular, tal como distancia focal y desplazamiento de punto principal en la dirección horizontal, y una ubicación de cámara horizontal del mundo real para cada visualización.

[0045] De esta manera, el dispositivo de destino 16 puede sintetizar visualizaciones utilizando información de textura y profundidad para una visualización particular, por ejemplo, para lograr una visualización sintetizada en un punto de visualización particular para lograr un efecto particular de renderización 3D (por ejemplo, para crear relativamente más o menos profundidad). Por ejemplo, para crear una profundidad percibida relativamente mayor, el dispositivo de destino 16 puede sintetizar una visualización que tiene una posición de cámara que está relativamente más lejos de la posición de cámara de la visualización codificada, mientras que para crear una profundidad percibida relativamente menor, el dispositivo de destino 16 puede sintetizar una visualización que tiene una posición de la cámara que está relativamente más cerca de la posición de la cámara de la visualización codificada. De esta manera, el dispositivo de destino 16 puede configurarse para usar parámetros de cámara, tales como ubicaciones de cámaras del mundo real para visualizaciones codificadas del flujo de bits, para sintetizar una visualización utilizada para producir un efecto tridimensional cuando la visualización del flujo de bits se visualiza junto con la visualización sintetizada.

[0046] El flujo de bits codificado 8 puede modularse mediante el módem 23 de acuerdo con una norma de comunicación, por ejemplo, como acceso múltiple por división de código (CDMA) u otra norma de comunicación, y transmitirse al dispositivo de destino 16 a través del transmisor 24 y el canal de comunicación 15. El módem 23 puede incluir diversos mezcladores, filtros, amplificadores u otros componentes diseñados para la modulación de señales. El transmisor 24 puede incluir circuitos diseñados para transmitir datos, incluidos amplificadores, filtros y una o más antenas. En algunos ejemplos, en lugar de transmitir a través de un canal de comunicación, el dispositivo de origen 12 almacena datos de vídeo codificados (por ejemplo, flujo de bits codificado 8) en un medio de almacenamiento, como un disco de vídeo digital (DVD), disco Blu-ray, memoria flash o similares.

[0047] El receptor 26 del dispositivo de destino 16 recibe información a través del canal de comunicación 15, y el módem 27 desmodula la información. Como el transmisor 24, el receptor 26 puede incluir circuitos diseñados para recibir datos, que incluyen amplificadores, filtros y una o más antenas. En algunos casos, el transmisor 24 y/o el receptor 26 pueden incorporarse dentro de un único componente de transceptor que incluye tanto circuitería de recepción como de transmisión. El módem 27 puede incluir varios mezcladores, filtros, amplificadores u otros componentes diseñados para la desmodulación de señales. En algunos casos, los módems 23 y 27 pueden incluir componentes para realizar la modulación y la desmodulación.

[0048] El término "codificador" se utiliza en el presente documento para referirse a un dispositivo o aparato informático especializado que realiza la codificación o la descodificación de vídeo. El término "codificador" se refiere en general a cualquier codificador de vídeo, descodificador de vídeo o codificador/descodificador combinado (códec). El término "codificación" se refiere a la codificación o la descodificación. El término "datos de vídeo codificados" puede referirse a cualquier unidad descodificable independientemente de datos de vídeo tales como un bloque, una trama completa, un fragmento de una trama, un bloque de datos de vídeo u otra unidad descodificable independientemente definida de acuerdo con las técnicas de codificación usadas.

[0049] El dispositivo de visualización 30 muestra los datos de vídeo descodificados 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 visualizaciones 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 visualizaciones simultáneamente, y las gafas pasivas pueden incluir lentes polarizadas que en general están polarizadas en direcciones ortogonales para filtrar entre las diferentes visualizaciones.

[0050] En el ejemplo de la FIG. 1, el enlace de comunicación 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 canal de comunicación 15 puede formar parte de una red basándose en paquetes, tal como una red de área local, una red de área extensa o una red global tal como Internet. El canal de comunicación 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 canal de comunicación 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 al dispositivo de destino 16.

[0051] El codificador de vídeo 22 y el descodificador de vídeo 28 pueden funcionar de acuerdo con una norma de compresión de vídeo, tal como la norma ITU-T H.264, descrita de forma alternativa como MPEG-4, Parte 10, Codificación de Vídeo Avanzada (AVC). Otras normas de compresión de vídeo, que están basadas en la norma ITU H.264/AVC y que pueden ser utilizadas por el codificador de vídeo 22 y el descodificador de vídeo 28, incluyen la norma de codificación de vídeo escalable (SVC), que es una extensión escalable para la norma ITU H.264/AVC. Otra norma con el codificador de vídeo 22 y el descodificador de vídeo 28 puede funcionar de acuerdo con la norma de codificación de vídeo de múltiples visualizaciones (MVC), que es una extensión de múltiples visualizaciones a la norma ITU H.264/AVC. Sin embargo, las técnicas de la presente divulgación no están limitadas a ninguna norma de codificación de vídeo particular.

[0052] En algunos aspectos, cada uno de entre el codificador de vídeo 22 y el descodificador de vídeo 28 puede estar integrado con un codificador y descodificador 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 multiplexación ITU H.223 o a otros protocolos, tales como el protocolo de datagramas de usuario (UDP).

[0053] El codificador de vídeo 22 y el descodificador 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 por campo (FPGA), lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. 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 las instrucciones. El codificador de vídeo 22 y el descodificador de vídeo 28 pueden incluirse cada uno en uno o más codificadores o descodificadores, donde cualquiera de los cuales puede estar integrado como parte de un códec combinado que proporciona capacidades de codificación y descodificación en un respectivo dispositivo móvil, dispositivo de abonado, dispositivo de radiodifusión, servidor o similares.

[0054] 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 ITU-T H.264, por ejemplo, cada fragmento puede incluir una serie de macrobloques, que pueden estar dispuestos en subbloques. La norma H.264 soporta 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 8x8 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 extenderse a flujos de vídeo 3D que incluyen dos componentes, textura y profundidad.

[0055] Los bloques de vídeo más pequeños pueden proporcionar una mejor resolución y pueden usarse en ubicaciones de una trama de vídeo que incluyan altos niveles de detalle. En general, los macrobloques y los diversos subbloques pueden considerarse como bloques de vídeo. Además, un fragmento de datos de vídeo puede considerarse una serie de bloques de vídeo, como macrobloques y/o subbloques. Cada fragmento puede ser una unidad independientemente descodificable de una trama de vídeo. De forma alternativa, las propias tramas pueden ser unidades descodificables, o pueden definirse otras partes de una trama como unidades descodificables.

[0056] Las unidades de datos de vídeo 2D (por ejemplo, tramas o fragmentos) de la norma ITU-T H.264 pueden extenderse a 3D codificando la información de profundidad de codificación desde un mapa de profundidad o mapa de paralaje además de los componentes de textura para esa trama o fragmento de vídeo. La asignación por paralaje (también conocido como asignación por desplazamiento virtual o asignación de desplazamiento) desplaza componentes de textura situados en una ubicación de píxel basándose en una función de un ángulo de visualización y un mapa de altura de la ubicación de píxel. El codificador de vídeo 22 puede codificar la información de profundidad como vídeo monocromático. Esta información de profundidad de vídeo codificada monocromática para una visualización puede codificarse y encapsularse en una unidad de vídeo (por ejemplo, flujo de bits) junto con el componente de textura de la misma visualización. De acuerdo con las técnicas de esta divulgación, un componente de visualización de una visualización de datos de vídeo puede incluir tanto componentes de textura como de profundidad codificados y encapsulados juntos en el mismo flujo de bits. El componente de visualización es una unidad de datos que incluye todos los datos para una instancia temporal común para una visualización particular (por ejemplo, una cámara o un elemento de captura de vídeo particular).

[0057] Para codificar bloques de vídeo, tales como bloques de vídeo que constituyen los componentes del flujo de bits codificado 8, el codificador de vídeo 22 realiza una predicción intra o predicción inter para generar uno o más bloques de predicción. El codificador de vídeo 22 puede generar bloques de predicción para la unidad de datos de componentes de textura (por ejemplo, trama o fragmento) y la unidad de datos de componentes de profundidad. Además, el codificador de vídeo 22 puede realizar predicción intra o entre visualizaciones para generar uno o más bloques de predicción para una visualización, donde la predicción intra-visualización puede explotar la dependencia entre componentes dentro de la misma visualización y la predicción entre visualizaciones puede explotar la dependencia entre componentes de diferentes visualizaciones.

[0058] 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 usados para representar los coeficientes. Después de la cuantificación, la codificación por entropía se puede realizar de acuerdo con una metodología de codificación por 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.

[0059] 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 esfuerzos realizados en la normalización se basan en un modelo de un dispositivo de codificación de vídeo denominado Modelo de Prueba de HEVC (HM). El HM supone varias capacidades de los dispositivos de codificación de vídeo respecto a dispositivos de acuerdo con, por ejemplo, la norma ITU-T H.264/AVC. Por ejemplo, mientras que la norma H.264 proporciona nueve modos de codificación de predicción intra, el HM proporciona hasta treinta y cuatro modos de codificación de predicción intra. HEVC puede extenderse para soportar las dos unidades de vídeo componentes como se describe en el presente documento.

[0060] El HM se refiere a un bloque de datos de vídeo como 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. El bloque codificado de dos componentes (es decir, componentes de textura y profundidad) puede ser una CU de acuerdo con la norma HM. Por lo tanto, una CU puede dividirse en subCU. 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 subCU de una LCU. Una LCU puede dividirse en varias subCU, y cada subCU puede dividirse en varias subCU. Los datos sintácticos para un flujo de bits pueden definir un número máximo de veces en que puede dividirse una LCU, denominada profundidad de CU. Por consiguiente, un flujo de bits también puede definir una unidad de codificación de mínimo tamaño (SCU). La presente divulgación también utiliza el término «bloque» para referirse a cualquiera de entre una CU, una unidad de predicción (PU) o una unidad de transformada (TU).

[0061] Una LCU puede asociarse a una estructura de datos de árbol cuádruple. En general, una estructura de datos en árbol cuádruple incluye un nodo por CU, donde un nodo raíz corresponde a la LCU. Si una CU se divide en 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 cuádruple puede proporcionar datos sintácticos para la CU correspondiente. Por ejemplo, un nodo en el árbol cuádruple 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 en varias sub-CU.

[0062] Una CU que no esté 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 de manera intramodal, la PU puede incluir datos que describan un modo de predicción intra para la PU. Como otro ejemplo, cuando la PU está codificada de modo de inter, la PU puede incluir datos que definen un vector de movimiento para la PU. Los datos que definan 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 con una precisión de un octavo de píxel), una trama de referencia a la que apunte 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 se puede tratar con diferentes resoluciones para los componentes de textura 4 y la información de profundidad 6. Los datos para la CU que definan la(s) PU también pueden describir, por ejemplo, una partición de la CU en una o más PU. Los modos de partición pueden diferir en función de si la CU está sin codificar, codificada en el modo de predicción intra o codificada en el modo de predicción inter.

[0063] 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 al tamaño de la CU correspondiente.

[0064] 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 un modo de predicción intra particular. Un codificador HM puede configurarse con hasta treinta y cuatro modos de predicción intra. Por lo tanto, para soportar una asignación de uno a uno entre los modos de predicción intra direccional y las transformadas direccionales, los codificadores y descodificadores HM necesitarían almacenar 66 matrices para cada tamaño de transformada soportado. Además, los tamaños de bloque para los cuales se soportan los treinta y cuatro modos de predicción intra pueden ser bloques relativamente grandes, por ejemplo, de 32x32 píxeles, 64x64 píxeles o incluso de mayor tamaño.

[0065] Después de codificar cada uno de los componentes de textura y profundidad, el codificador de vídeo 22 puede determinar cualquier dependencia de predicción entre los componentes de textura y profundidad de una visualización y/o entre los componentes de textura y profundidad de diferentes visualizaciones. El dispositivo de origen 12 puede implementar las técnicas de esta divulgación encapsulando componentes de textura y profundidad en las respectivas unidades VLC NAL correspondientes a la misma instancia temporal para una visualización, que puede tratarse como un componente de visualización común. Las técnicas de esta divulgación pueden proporcionar la codificación de datos de textura y datos de mapas de profundidad para una visualización consecutivamente en el flujo de bits (por ejemplo, flujo de bits codificado 8). Para separar la información de mapa de profundidad y textura de un componente de visualización, el codificador de vídeo 22 puede agregar un delimitador en el flujo de bits, por ejemplo, un delimitador de unidad NAL que indica el final de la información de textura y el comienzo de la información de profundidad para el componente de visualización. Adicionalmente, el codificador de vídeo 22 puede señalar dentro de las dependencias y parámetros de predicción intra y entre visualizaciones de flujo de bits asociados con la predicción de síntesis de visualización.

- 5 **[0066]** De acuerdo con otro aspecto de esta divulgación, el codificador de vídeo 22 puede señalar los parámetros de la cámara en los datos de vídeo y puede describir adicionalmente el valor de profundidad a la asignación de valores Z del mundo real. Los datos de vídeo también pueden describir una relación de predicción de visualizaciones de mapas de profundidad y visualizaciones de textura. El codificador de vídeo 22 puede indicar estos parámetros e información adicionales a nivel de secuencia (por ejemplo, en una estructura de datos de conjunto de parámetros de secuencia (SPS)), a nivel de imagen (por ejemplo, en una estructura de datos de conjunto de parámetros de imagen (PPS) o cabecera de trama), a nivel de fragmento (por ejemplo, en una cabecera de fragmento) o a nivel de bloque (por ejemplo, en una cabecera de bloque).
- 10 **[0067]** En el dispositivo de destino 16, el descodificador 28 de vídeo recibe los datos de vídeo codificados (por ejemplo, flujo de bits codificado 8). El descodificador de vídeo 28 puede diferenciar unidades NAL de textura (por ejemplo, el componente de textura) de unidades NAL de profundidad (por ejemplo, el componente de profundidad) usando un indicador que indica si una unidad NAL incluye información de textura o profundidad, como se analiza a continuación. Después de descodificar los componentes de profundidad, el dispositivo de destino 16 puede usar una visualización 2D (es decir, información de textura) y descartar otras visualizaciones e información de profundidad; use una visualización 2D y su información de profundidad correspondiente para sintetizar una visualización virtual para reproducción en 3D; use dos o más visualizaciones 2D para la reproducción en 3D; o use múltiples visualizaciones y su información de profundidad para producir múltiples visualizaciones virtuales para la reproducción de múltiples visualizaciones. La determinación de cómo reproducir el contenido puede depender de factores tales como, por ejemplo, el tipo de dispositivo de visualización del dispositivo 30 y los formatos de vídeo soportados por el dispositivo de visualización 30.
- 15 **[0068]** Después de separar los componentes de textura y profundidad del flujo de bits codificado 8, la entropía del descodificador de vídeo 28 descodifica los datos de vídeo recibidos, tales como bloques codificados correspondientes a cada uno de los componentes de acuerdo con una metodología de codificación por entropía, como CAVLC o CABAC, para obtener los coeficientes cuantificados. El descodificador de vídeo 28 aplica las funciones de cuantificación inversa (descuantificación) y de transformada inversa para reconstruir el bloque residual en el dominio del píxel. El descodificador 28 de vídeo genera también un bloque de predicción basado en información de control o información de sintaxis (por ejemplo, modo de codificación, vectores de movimiento, sintaxis que define coeficientes de filtro y similares) incluidos en los datos de vídeo codificados. El descodificador 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 descodificación realizado por el descodificador de vídeo 28, con respecto a la FIG. 7.
- 20 **[0069]** Como se describe en el presente documento, la información de textura puede comprender tres componentes; por ejemplo, Y puede representar luminancia, Cb y Cr pueden representar dos valores diferentes de crominancia de un espacio de color YCbCr tridimensional (por ejemplo, matices azul y rojo), y D puede representar información de profundidad. En algunos ejemplos, cada ubicación de píxel puede realmente definir un valor de píxel para la textura del píxel y un valor de píxel para la profundidad de la ubicación del píxel. El valor de textura puede comprender, por ejemplo, dos o más valores que representan componentes de luminancia y crominancia. Además, los componentes de profundidad y textura pueden tener resoluciones diferentes. Tanto el componente de profundidad como el componente de textura pueden dividirse en bloques de píxeles respectivos, cada uno de los cuales puede codificarse individualmente.
- 25 **[0070]** En algunos ejemplos, el codificador de vídeo 22 y el descodificador 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 descodificador de vídeo 28 pueden aplicar un filtro de interpolación al soporte de filtro que comprende conjuntos de posiciones de píxel entero.
- 30 **[0071]** Como se indicó anteriormente, el descodificador de vídeo 28 del dispositivo de destino 16 puede recibir el flujo de bits codificado 8, que puede incluir componentes de textura y profundidad, junto con información adicional, incluyendo dependencias de predicción e información con respecto a la visualización asociada con los componentes de textura y profundidad. Además, los parámetros de la cámara pueden señalarse para la visualización. El descodificador de vídeo 28 puede renderizar datos de vídeo para la reproducción en 3D basándose en los componentes del flujo de bits codificado 8 y una o más información adicional que incluye información de visualización y parámetros de la cámara. De acuerdo con las técnicas de esta divulgación, y como se analiza en mayor detalle a continuación, la información incluida para una o más visualizaciones puede utilizarse para predecir el componente de textura de una visualización a partir de componentes de textura de una o más visualizaciones diferentes, y el componente de profundidad de una visualización basado en el componente de textura de la misma visualización y/o componentes de profundidad de una o más visualizaciones diferentes.
- 35 **[0072]** 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 puede codificar unidades de bloque asociadas con imágenes correspondientes a diferentes visualizaciones proporcionadas por una fuente de vídeo, donde cada visualización puede comprender un componente de textura y un componente de profundidad. 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
- 40 **[0069]** Como se describe en el presente documento, la información de textura puede comprender tres componentes; por ejemplo, Y puede representar luminancia, Cb y Cr pueden representar dos valores diferentes de crominancia de un espacio de color YCbCr tridimensional (por ejemplo, matices azul y rojo), y D puede representar información de profundidad. En algunos ejemplos, cada ubicación de píxel puede realmente definir un valor de píxel para la textura del píxel y un valor de píxel para la profundidad de la ubicación del píxel. El valor de textura puede comprender, por ejemplo, dos o más valores que representan componentes de luminancia y crominancia. Además, los componentes de profundidad y textura pueden tener resoluciones diferentes. Tanto el componente de profundidad como el componente de textura pueden dividirse en bloques de píxeles respectivos, cada uno de los cuales puede codificarse individualmente.
- 45 **[0070]** En algunos ejemplos, el codificador de vídeo 22 y el descodificador 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 descodificador de vídeo 28 pueden aplicar un filtro de interpolación al soporte de filtro que comprende conjuntos de posiciones de píxel entero.
- 50 **[0071]** Como se indicó anteriormente, el descodificador de vídeo 28 del dispositivo de destino 16 puede recibir el flujo de bits codificado 8, que puede incluir componentes de textura y profundidad, junto con información adicional, incluyendo dependencias de predicción e información con respecto a la visualización asociada con los componentes de textura y profundidad. Además, los parámetros de la cámara pueden señalarse para la visualización. El descodificador de vídeo 28 puede renderizar datos de vídeo para la reproducción en 3D basándose en los componentes del flujo de bits codificado 8 y una o más información adicional que incluye información de visualización y parámetros de la cámara. De acuerdo con las técnicas de esta divulgación, y como se analiza en mayor detalle a continuación, la información incluida para una o más visualizaciones puede utilizarse para predecir el componente de textura de una visualización a partir de componentes de textura de una o más visualizaciones diferentes, y el componente de profundidad de una visualización basado en el componente de textura de la misma visualización y/o componentes de profundidad de una o más visualizaciones diferentes.
- 55 **[0072]** 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 puede codificar unidades de bloque asociadas con imágenes correspondientes a diferentes visualizaciones proporcionadas por una fuente de vídeo, donde cada visualización puede comprender un componente de textura y un componente de profundidad. 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
- 60

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/descodificador (CÓDEC)) también pueden realizar técnicas similares a las realizadas por el codificador de vídeo 22.

5 **[0073]** El codificador de vídeo 22 puede realizar intracodificación e intercodificación de bloques dentro de tramas de vídeo, aunque los componentes de intracodificación no se muestran en la FIG. 2 para facilitar la ilustración. La intracodificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial en el vídeo en una trama de vídeo dada. La intercodificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal en el vídeo dentro de tramas adyacentes de una secuencia de vídeo. El modo intra (modo I) puede referirse al modo de compresión espacial. Los modos inter, tales como una predicción unidireccional (modo P) o una predicción bidireccional (modo B), pueden referirse a los modos de compresión temporal. 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.

15 **[0074]** Como se muestra en la FIG. 2, el codificador de vídeo 22 recibe bloques de vídeo asociados con componentes de textura y profundidad de una trama de vídeo a codificar, donde la trama de vídeo puede corresponder a una de las múltiples visualizaciones en un momento específico. Por ejemplo, el codificador de vídeo 22 puede recibir componentes de textura 4 y componentes de profundidad 6. En el ejemplo de la FIG. 2, el codificador de vídeo 22 incluye una unidad de codificación de predicción (MCU) 32, una memoria 34, un sumador 48, una unidad de transformada 38, una unidad de cuantificación 40 y una unidad de codificación por entropía 46. Para la reconstrucción del bloque de vídeo, el codificador de vídeo 22 también incluye una unidad de cuantificación inversa 42, una unidad de transformada inversa 44, un sumador 51 y una unidad de desbloqueo 43. La unidad de desbloqueo 43 puede ser un filtro de desbloqueo que puede filtrar los límites del bloque para eliminar distorsiones de bloqueo del vídeo reconstruido. Si está incluida en el codificador de vídeo 22, la unidad de desbloqueo 43 típicamente filtraría la salida del sumador 51. En un ejemplo, como se muestra en la FIG. 2, la unidad de transformada 38 puede ser un bloque funcional, no una "TU" en términos de HEVC. La unidad de desbloqueo 43 puede determinar información de desbloqueo para el uno o más componentes de textura. La unidad de desbloqueo 43 también puede determinar información de desbloqueo para el componente de mapa de profundidad. En algunos ejemplos, la información de desbloqueo para el uno o más componentes de textura puede ser diferente de la información de desbloqueo para el componente de mapa de profundidad.

25 **[0075]** En algunos ejemplos, el codificador de vídeo 22 puede incluir una unidad de síntesis de visualización para sintetizar visualizaciones virtuales usadas como referencia cuando se codifican datos de otras visualizaciones. La unidad de síntesis de visualización no se muestra en la ilustración de la FIG. 2 para mayor claridad, pero cuando se proporcione, normalmente recuperará datos de la memoria 34 (como información de textura y profundidad para un componente de visualización particular), sintetizará un componente de visualización incluyendo información de textura usando los datos recuperados, y almacenará el componente de visualización sintetizado en la memoria 34 para usar como referencia al codificar datos de vídeo de otras visualizaciones.

35 **[0076]** La unidad de predicción 32 recibe el bloque de vídeo (etiquetado "BLOQUE DE VÍDEO" en la FIG. 2) que comprende componentes de textura 4 o información de profundidad 6. La unidad de predicción 32 puede incluir una unidad de estimación de movimiento (ME) 35 y una unidad de compensación de movimiento (MC) 37. La unidad de predicción 32 puede predecir la información de profundidad en los bloques de componentes de profundidad y la información de textura en los bloques de componentes de textura. Uno o más filtros de interpolación 39 (denominados "filtro 39" en el presente documento) pueden estar incluidos en la unidad 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 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 predicción 32 puede incluir una pluralidad de filtros de interpolación o de filtros tipo interpolación. La unidad de predicción 32 puede predecir información de profundidad y/o textura usando información de profundidad y/o textura desde la misma visualización y/u otras visualizaciones asociadas con una instancia temporal común. La unidad de predicción 32 también puede realizar la predicción de síntesis de visualización, que puede proporcionar información de predicción que permite la predicción de un bloque de una visualización con respecto a los datos de una visualización sintetizada.

40 **[0077]** Durante el proceso de codificación, el codificador de vídeo 22 recibe un bloque de vídeo para codificar (designado "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 vídeo a codificar y el bloque de predicción pueden corresponder a componentes de textura 4 e información de profundidad 6. 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 bloque de predicción. La unidad ME 35 y la unidad MC 37 pueden realizar una estimación de movimiento y una compensación de movimiento, respectivamente, usando bloques correspondientes a componentes de textura y/o profundidad desde la misma visualización u otras visualizaciones asociadas con una instancia temporal común. La unidad ME 35 puede de hecho producir un "vector de desplazamiento" cuando realiza una predicción entre visualizaciones, que es conceptualmente similar a un vector de movimiento, pero en general predice el desplazamiento

horizontal de un bloque particular debido a diferencias en posiciones horizontales de las cámaras para las visualizaciones respectivas, en lugar de movimiento de objetos a lo largo del tiempo. Por lo tanto, la unidad 37 de MC también puede denominarse "unidad de compensación de movimiento/desplazamiento". De esta manera, la unidad ME 35 y la unidad MC 37 pueden hacer determinaciones que se pueden usar en la predicción entre visualizaciones e intra-visualización de componentes de textura y profundidad.

[0078] 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 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 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.

[0079] La unidad ME 35 calcula al menos un vector de movimiento para el bloque de vídeo que se codifica comparando el bloque de vídeo con bloques de referencia de una o más tramas de referencia (por ejemplo, una trama o tramas previas y/o posteriores de otras visualizaciones en la misma instancia temporal). Los datos para las tramas de referencia (por ejemplo, componentes de textura y/o profundidad) 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, del fraccionario, subentero o subpíxel. En la estimación de movimiento de píxel fraccionario, la unidad de ME 35 puede calcular un vector de movimiento que indica un desplazamiento a una ubicación distinta de una ubicación de píxel entera. 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 píxeles fraccionarios permite a la unidad de predicción 32 estimar el movimiento con mayor precisión que las ubicaciones de píxel entero (o píxel completo) y así, la unidad de predicción 32 puede generar un bloque de predicción más preciso.

[0080] 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 filtro(s) 39 a bloques reconstruidos que se almacenarán en la memoria 34.

[0081] Para realizar una compensación de movimiento de píxel fraccionario, la unidad MC 37 puede realizar una interpolación (a veces denominada filtrado de interpolación) para generar datos a una resolución de subpíxeles (denominada en el presente documento como valores de subpíxeles o de píxel fraccionario). La unidad MC 37 puede recurrir a uno o más filtros 39 para esta interpolación. De esta manera, la unidad de predicción 32 puede calcular valores para posiciones de píxeles subenteros de un bloque de referencia, que luego puede usarse para producir un bloque predictivo de datos de vídeo para cada uno de los componentes de profundidad y textura de la unidad de bloque 3D. En un ejemplo, se puede producir un bloque predictivo para el componente de textura de la unidad de bloque 3D, y el bloque predictivo o el componente de textura de la unidad de bloque 3D se puede utilizar para producir el componente de profundidad de la misma unidad de bloque 3D, correspondiente a la misma visualización u otras visualizaciones asociadas con una instancia temporal común.

[0082] Una vez calculado el vector de movimiento para el bloque de vídeo a codificar por la unidad ME 35, la unidad MC 37 genera el bloque de vídeo de predicción asociado con ese vector de movimiento. La unidad MC 37 puede extraer el bloque de predicción de la memoria 34 basándose en el vector de movimiento determinado por la unidad MC 35. En el caso de un vector de movimiento con precisión de píxel fraccionario, la unidad MC 37 puede filtrar los datos de la memoria 34 para interpolar dichos datos a una resolución de subpíxeles, por ejemplo, recurriendo a uno o más filtros 39 para este proceso.

[0083] En algunos casos, las posiciones de píxeles de entero completo correspondientes a los coeficientes de filtro utilizados para interpolar tales datos a la resolución de subpíxeles pueden indicarse como uno o más elementos sintácticos de interpolación a la unidad de codificación por entropía 46 para su inclusión en el flujo de bits codificado. De manera similar, la técnica o modo de filtrado de interpolación que se usó para generar los datos de predicción de subpíxel también puede indicarse como uno o más elementos sintácticos de interpolación a la unidad de codificación por entropía 46 para su inclusión en el flujo de bits codificado.

[0084] Una vez que la unidad de predicción 32 ha generado el bloque de predicción, el codificador de vídeo 22 forma un bloque de vídeo residual (denominado "BLOQUE RESID." en la FIG. 2) restando el bloque de predicción del bloque de vídeo original que se está codificando. Esta resta puede ocurrir entre componentes de textura en el bloque de vídeo original y componentes de textura en el bloque de predicción de textura, 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 de profundidad. El sumador 48 representa el componente o los componentes que realizan esta operación de resta.

5 **[0085]** La unidad de transformada 38 aplica una transformada, tal como una transformada discreta de coseno (DCT) o una transformada conceptualmente similar, al bloque residual, produciendo un bloque de vídeo que comprende coeficientes de bloque de transformada residual. Debe entenderse que la unidad 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 transformada 38 puede realizar, por ejemplo, otras transformadas, tales como las definidas por la norma H.264, que sean conceptualmente similares a la DCT. Dichas transformadas pueden incluir, 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 transformada 38 aplica la transformada al bloque residual, produciendo un bloque de coeficientes de transformada residuales. La unidad 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 puede convertir la información residual de un dominio de píxel en un dominio de frecuencia.

20 **[0086]** La unidad de cuantificación 40 cuantifica los coeficientes de transformada residuales para reducir más la velocidad de transmisión de bits. El proceso de cuantificación puede reducir la profundidad de bits asociada con algunos o la totalidad de los coeficientes. La unidad de cuantificación 40 puede cuantificar un residuo de codificación de imagen de profundidad. Tras la cuantificación, la unidad de codificación por entropía 46 realiza la codificación por entropía de los coeficientes de transformada cuantificados. Por ejemplo, la unidad de codificación por entropía 46 puede realizar la CAVLC, la CABAC u otro procedimiento de codificación por entropía.

25 **[0087]** La unidad de codificación por entropía 46 puede codificar también uno o más vectores de movimiento e información de soporte obtenida de la unidad 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 un modo 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 desbloqueo para la imagen de profundidad u otra información asociada con la generación del bloque de predicción. Estos elementos sintácticos de predicción pueden proporcionarse en el nivel de secuencia o en el nivel de imagen. De acuerdo con las técnicas de esta divulgación, la unidad de codificación por entropía 46 también puede codificar información con respecto a las dependencias de predicción de intra y entre visualizaciones y los parámetros asociados con la predicción de la síntesis de visualización.

40 **[0088]** El uno o más elementos sintácticos pueden incluir también una diferencia de parámetro de cuantificación (QP) entre el componente de textura y el componente de profundidad. La diferencia de QP se puede señalar a nivel de fragmento. Otros elementos sintácticos también pueden señalarse a un nivel de unidad de bloque codificada, que incluye un patrón de bloque codificado para la información de profundidad, un QP delta para la información 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).

50 **[0089]** De acuerdo con las técnicas de esta divulgación, después de la codificación por entropía mediante la unidad de codificación por entropía 46, el componente de textura y el componente de profundidad de una visualización pueden encapsularse en unidades VCL NAL correspondientes a la misma instancia temporal para la visualización. La unidad de codificación por entropía 46 puede añadir una cabecera de unidad NAL a los respectivos fragmentos y tramas de información de textura o profundidad para encapsular información de profundidad o textura dentro de una unidad NAL respectiva. El codificador de vídeo 22 también puede agregar un delimitador de unidad NAL en el flujo de bits que indica el final de los datos de textura y el comienzo de la información de profundidad para la visualización. De forma alternativa, el codificador de vídeo 22 puede establecer un valor para un indicador de profundidad incluido en la cabecera de la unidad NAL para indicar si la unidad NAL incluye información de textura o profundidad.

60 **[0090]** Adicionalmente, las técnicas de esta divulgación pueden proporcionar una señal dentro del flujo de bits con respecto a las dependencias de predicción intra y entre visualizaciones y los parámetros asociados con la predicción de síntesis de visualización. Es decir, el codificador de vídeo 22 puede proporcionar información indicativa de elementos particulares usados como referencia para componentes de textura y componentes de profundidad. El flujo de bits codificado que encapsula los componentes de textura y profundidad de una visualización puede luego transmitirse a otro dispositivo o archivar (por ejemplo, en la memoria 34) para su posterior transmisión o recuperación.

[0091] La unidad de cuantificación inversa 42 y la unidad de transformada inversa 44 aplican la cuantificación inversa y la transformada inversa, respectivamente, para reconstruir el bloque residual en el dominio de píxeles, por ejemplo, para su uso posterior como un bloque de referencia. El bloque residual reconstruido (denominado "BLOQUE RESID. RECON." en la FIG. 2) puede representar una versión reconstruida del bloque residual proporcionado para la unidad 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 predicción 32, para generar un bloque de vídeo reconstruido para su almacenamiento en la memoria 34. El bloque de vídeo reconstruido puede ser utilizado por la unidad de predicción 32 como bloque de referencia que se puede utilizar para codificar posteriormente una unidad de bloque en una trama de vídeo posterior o una unidad codificada posterior.

[0092] De esta manera, el codificador de vídeo 22 representa un ejemplo de un codificador de vídeo configurado para recibir datos de textura para una instancia temporal de una visualización de datos de vídeo, recibir datos de profundidad correspondientes a los datos de textura para la instancia temporal de la visualización de datos de vídeo, y encapsular los datos de textura y los datos de profundidad en un componente de visualización para la instancia temporal de la visualización, de manera que los datos de textura y los datos de profundidad se encapsulan dentro de un flujo de bits común. El codificador de vídeo 22 también puede señalar información con respecto a dependencias de predicción tales como, por ejemplo, predicción intra (por ejemplo, predicción de información de profundidad a partir de otra información de profundidad de la misma visualización), predicción temporal o entre visualizaciones (por ejemplo, predicción de información de profundidad a partir de información de profundidad de un componente de visualización diferente o de la información de textura de un componente de visualización diferente), y los parámetros de cámara para una cámara utilizada para capturar los datos de textura de la visualización de datos de vídeo.

[0093] Las FIGs. 3A-3B son diagramas conceptuales que ilustran ejemplos de elementos de una estructura de flujo de bits que pueden usarse en una implementación de una o más de las técnicas de esta divulgación. Se puede usar un flujo de bits para transferir unidades de bloque MVD de dos componentes y elementos sintácticos entre, por ejemplo, el dispositivo de origen 12 y el dispositivo 16 de destino de la FIG. 1. El flujo de bits puede cumplir con la norma de codificación ITU H.264/AVC y, en particular, seguir una estructura de flujo de bits MVC. Es decir, el flujo de bits puede ajustarse a la extensión MVC de H.264/AVC, en algunos ejemplos. En otros ejemplos, el flujo de bits se puede ajustar a una extensión de múltiples visualizaciones de la HEVC o una extensión de múltiples visualizaciones de otra norma. En otros ejemplos más, se pueden usar otras normas de codificación.

[0094] Una disposición de orden (orden de descodificación) 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 visualizaciones para un instante de tiempo de salida. El orden de descodificación de las unidades de acceso puede o no ser idéntico al orden de salida o de visualización. Típicamente, la predicción de MVC puede incluir tanto la predicción inter de imágenes dentro de cada visualización como la predicción inter de visualizaciones. En la MVC, la predicción inter de visualizaciones puede soportarse 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 visualización diferente como imagen de referencia.

[0095] La MVC soporta la codificación de dos visualizaciones. Una de las ventajas de la MVC es que un codificador MVC puede tomar más de dos visualizaciones como entrada de vídeo 3D, y un decodificador MVC puede descodificar las dos visualizaciones como una representación de múltiples visualizaciones. Por lo tanto, un renderizador con un decodificador MVC puede tratar un contenido de vídeo 3D como si tuviera múltiples visualizaciones. Anteriormente, MVC no procesaba la entrada del mapa de profundidad, similar a H.264/AVC con mensajes SEI (imágenes de intercalado espacial o información estéreo).

[0096] En la norma H.264/AVC, se definen unidades de capa de abstracción de red (NAL) para proporcionar una representación de vídeo adecuada para la red destinada a aplicaciones como la videotelefonía, el almacenamiento o la reproducción de vídeo. Las unidades NAL pueden clasificarse en unidades NAL de la capa de codificación de vídeo (VCL) y unidades NAL no de la 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 de la VCL.

[0097] Para la codificación de vídeo 2D, cada unidad NAL puede contener una cabecera de unidad NAL de un byte y una carga útil de tamaño variable. Se pueden usar cinco bits para especificar el tipo de unidad NAL. Se pueden utilizar 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 puede ampliar 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 en la cabecera de unidad NAL se pueden utilizar para indicar que la unidad NAL es una unidad NAL de cuatro componentes.

[0098] Las cabeceras de unidad NAL también se pueden utilizar como unidades MVC NAL. 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 MVC NAL pueden incluir `priority_id`, `temporal_id`, `anchor_pic_flag`, `view_id`, `non_idr_flag` e `inter_view_flag`. En otros ejemplos, otros elementos sintácticos se pueden incluir en una cabecera de unidad MVC NAL.

[0099] 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 descodificar correctamente sin descodificar las imágenes previas en el orden de descodificación (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.

[0100] 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 visualización. Este identificador de la cabecera de la unidad NAL permite la fácil identificación de unidades NAL en el descodificador y el acceso rápido de las visualizaciones descodificadas 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 visualización, pero pueden no depender de ninguna imagen codificada con un `temporal_id` superior.

[0101] 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 visualizaciones de otra unidad NAL en una visualización diferente.

[0102] 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 pueden contener información de cabecera a nivel de secuencia en conjuntos de parámetros de secuencia y la información de cabecera a nivel de imagen que cambia de manera poco frecuente en los conjuntos de parámetros de imagen (PPS). Con los conjuntos de parámetros, esta información que cambia con poca frecuencia no necesita repetirse para cada secuencia o imagen, por lo que se mejora la eficiencia de la 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 pueden transmitirse en un canal diferente al de las otras unidades NAL. En la MVC, una dependencia de visualización se puede señalar en la extensión MVC SPS. Todas las predicciones inter de visualizaciones pueden realizarse dentro del alcance especificado por la extensión MVC SPS.

[0103] En las técnicas de codificación de vídeo 3D, puede ser deseable codificar conjuntamente la textura y la profundidad del contenido de MVD con una alta eficiencia. Sin embargo, para lograr de manera eficiente la codificación conjunta de los componentes de textura y profundidad, puede ser deseable cumplir ciertos requisitos. Por ejemplo, puede ser preferible gestionar las imágenes codificadas y las imágenes descodificadas de la profundidad y la textura de forma global. Por lo tanto, puede ser deseable utilizar una unidad NAL distinta para el mapa de profundidad a partir de una unidad NAL para la textura dentro de una visualización. También puede ser deseable que el formato sea compatible con ciertas normas, por ejemplo, H.264/AVC y MVC. Además, puede haber correlaciones entre el mapa de profundidad y la textura de la misma visualización y entre diferentes visualizaciones, que pueden utilizarse para mejorar la eficiencia de codificación.

[0104] Como se describe en el presente documento, las técnicas se pueden agregar a normas existentes, tales como MVC, para soportar vídeo 3D. Se puede añadir vídeo de múltiples visualizaciones más profundidad (MVD) a la MVC para el procesamiento de vídeo 3D. Las técnicas de codificación de vídeo 3D pueden proporcionar más flexibilidad y extensibilidad a las normas de vídeo existentes, por ejemplo, para cambiar el ángulo de visualización sin problemas o ajustar la convergencia o la percepción de profundidad hacia atrás o hacia adelante basándose en las especificaciones de los dispositivos o las preferencias del usuario. Las normas de codificación también pueden ampliarse para utilizar mapas de profundidad para la generación de visualizaciones virtuales en vídeo 3D.

[0105] Las técnicas de esta divulgación pueden proporcionar el marco para la codificación de vídeo 3D. Por ejemplo, las imágenes codificadas y las imágenes descodificadas se pueden gestionar de manera óptima para textura y profundidad. Las técnicas de esta divulgación también pueden proporcionar una estructura que es compatible con el perfil estéreo MVC (por ejemplo, donde hay dos visualizaciones). Las técnicas también pueden proporcionar señalización en el flujo de bits de dependencias de predicción (por ejemplo, entre los componentes de textura y

profundidad de una visualización, o entre los componentes de textura de diferentes visualizaciones) y de parámetros beneficiosos para la codificación (por ejemplo, parámetros de cámara para cada visualización). De acuerdo con las técnicas de esta divulgación, se puede añadir un delimitador de unidad NAL en el flujo de bits, entre un componente de visualización de textura y un componente de visualización de profundidad. Por ejemplo, el delimitador puede indicar el final de los datos de textura (por ejemplo, componente de visualización de textura) y el comienzo de la información de profundidad (por ejemplo, componente de visualización de profundidad), el final de la información de profundidad o el comienzo de los datos de textura para el componente de visualización. Tal delimitador se puede usar, por ejemplo, en MPEG-2 TS u otros sistemas. En un ejemplo, el delimitador puede tener un tipo de unidad NAL diferente del delimitador de unidad NAL que se utiliza para separar los componentes de visualización de diferentes visualizaciones en MVC.

[0106] La FIG. 3A es un diagrama conceptual que ilustra un ejemplo de una estructura de flujo de bits 70 que puede usarse en una implementación de una o más de las técnicas de esta divulgación. En la FIG. 3A, el flujo de bits 70 cumple con la norma MVC después de ampliarse para incluir MVD, de acuerdo con las técnicas de esta divulgación. En otros ejemplos, el flujo de bits 70 puede ajustarse a otras normas de codificación de vídeo.

[0107] El flujo de bits 70 puede comprender información adicional relacionada con la señalización de las unidades de bloque. Por ejemplo, el flujo de bits 70 puede incluir una indicación de dependencias de predicción entre diferentes componentes, tales como la información de profundidad 6 con respecto a uno o más componentes de textura 4. En otros ejemplos, el flujo de bits 70 puede incluir una indicación de dependencias de predicción entre componentes de textura asociados con diferentes visualizaciones. En otros ejemplos, el flujo de bits 70 puede señalar parámetros de cámara para una visualización asociada con los componentes de textura y profundidad indicados en el flujo de bits.

[0108] Como se muestra en la FIG. 3A, el flujo de bits 70 comprende una pluralidad de unidades de acceso 72-1 a 72-N. Las unidades de acceso comprenden un conjunto de componentes de visualización (a los que se hace referencia como visualizaciones para mayor comodidad a continuación), tales como las visualizaciones 74-1 a 74-M. En general, las unidades de acceso incluyen todos los datos para una instancia temporal común, por ejemplo, datos para un componente de visualización por visualización. En algunos ejemplos, cada unidad de acceso 72-1 a 72-N comprende el mismo número de visualizaciones 74-1 a 74-M (denominada una visualización 74). La descodificación de cada unidad de acceso puede dar como resultado una imagen descodificada por visualización. Las unidades de acceso 72-1 a 72-N pueden contener datos de vídeo codificados que pueden usarse para renderizar vídeos 3D. De acuerdo con las técnicas de esta divulgación, los componentes de visualización de las unidades de acceso pueden incluir un componente de textura y uno de profundidad.

[0109] La FIG. 3B es un diagrama conceptual que ilustra un ejemplo de un componente de visualización 74-M que puede incluirse en la estructura del flujo de bits 70 de la FIG. 3A. Cada componente de visualización en una unidad de acceso (tal como el componente de visualización 74-1 en la unidad de acceso 72-1) contiene un conjunto de unidades NAL de capa de códec de vídeo (VCL) 76-1 a 76-3. El componente de visualización 74-M comprende las unidades NAL 76-1 a 76-3 en una forma y orden específicos. Típicamente, los componentes de visualización están dispuestos en el mismo orden en cada unidad de acceso, de manera que el k -ésimo componente de visualización en cada unidad de acceso corresponde a la misma visualización. En otros ejemplos, el componente de visualización 74-M comprende otros números de unidades NAL, algunas de las cuales pueden comprender información de textura, mientras que otras pueden comprender información de profundidad. De acuerdo con las técnicas de esta divulgación, las unidades NAL 76-1 a 76-3 pueden incluir información (por ejemplo, un indicador) que indica si la unidad NAL incluye información de textura o profundidad. Además, se puede utilizar un delimitador de unidad NAL para indicar dónde se detiene la unidad NAL de información de textura y comienza la unidad NAL de información de la profundidad.

[0110] En un ejemplo, al implementar la unidad NAL, se puede usar una extensión de cabecera de unidad NAL, donde se puede agregar un indicador en la cabecera de la unidad NAL para indicar si la unidad NAL actual contiene una unidad VCL NAL de un mapa de profundidad o la textura de una visualización. De forma alternativa, se puede agregar un nuevo tipo de unidad NAL para indicar si la unidad NAL es una unidad VCL NAL correspondiente a un fragmento de mapa de profundidad. Se podría agregar un bit en la unidad VCL NAL de un mapa de profundidad, para indicar si esta unidad VCL NAL se usa para predecir cualquier imagen codificada de la textura de una visualización. La semántica de `inter_view_flag` puede extenderse para proporcionar la indicación deseada, y si no, el `inter_view_flag` en la unidad VCL NAL de un mapa de profundidad indica que esta unidad NAL se usa para predecir el mapa de profundidad de otra visualización. En un ejemplo, una unidad NAL de prefijo puede contener la extensión de visualización base de la cabecera de la unidad NAL, `nal_unit_header_mvc_extension()`, definida de la siguiente manera:

```

nal_unit_header_mv_c_extension( ) {
    non_idr_flag
    priority_id
    view_id
    temporal_id
    anchor_pic_flag
    inter_view_flag
    depth_flag
    depth to view flag
}

```

donde depth_flag indica si el componente de visualización asociado es o no una unidad NAL de mapa de profundidad. El valor de depth_flag puede establecerse en 1 para indicar que la unidad NAL es una unidad VLC NAL codificada del mapa de profundidad, y se establece en 0 para indicar que la unidad NAL es una unidad VLC NAL codificada de la textura de una visualización. Cuando depth_to_view_flag es igual a 0, eso puede indicar que el componente de visualización actual no se usa para predecir componentes de visualización de profundidad, si es un componente de visualización de textura. Si se trata de un componente de visualización de profundidad, entonces no se usa para predecir componentes de textura. Cuando depth_to_view_flag es igual a 1, puede indicar que el componente de visualización actual puede usarse para predecir componentes de visualización de profundidad de la misma unidad de acceso, si es un componente de visualización de textura, y puede usarse para predecir componentes de visualización de textura de la misma unidad de acceso, si es un componente de visualización de profundidad. Cuando inter_view_flag es igual a 1, indica si el componente de visualización actual puede usarse para un componente de visualización, que junto con el componente de visualización actual son ambos un componente de visualización de textura o ambos un componente de visualización de profundidad.

[0111] La TABLA 1 a continuación ilustra ejemplos de códigos de tipo de unidad NAL, categorías de elementos sintácticos y clases de tipo de unidad NAL. nal_unit_type 21 puede ser introducido para las unidades NAL que contienen componentes de visualización de profundidad, concretamente slice_layer_depth_extension_rbsp().

TABLA 1

nal_unit_type	Contenido de la unidad NAL y estructura sintáctica Rbsp	C	Anexo A Clase de tipo de unidad NAL	Anexo G y anexo H Clase de tipo de unidad NAL
0	Sin especificar		no VCL	no VCL
1	Fragmento codificado de una imagen que no es IDR slice_layer_without_partitioning_rbsp()	2, 3, 4	VCL	VCL
...				
19	Fragmento codificado de una imagen codificada auxiliar sin dividir slice_layer_without_partitioning_rbsp()	2, 3, 4	no VCL	no VCL
20	Extensión de fragmento codificado slice_layer_extension_rbsp()	2, 3, 4	no VCL	VCL
21	Extensión de profundidad de fragmento codificado slice_layer_depth_extension_rbsp()	2, 3, 4	no VCL	VCL
21..23	Reservado		no VCL	no VCL
24..31	Sin especificar		no VCL	no VCL

[0112] En un ejemplo, se puede utilizar una extensión de profundidad de conjunto de parámetros de secuencia y puede tener el siguiente diseño general de ejemplo:

seq_parameter_set_depth_extension() {
depth_z_mapping_idc
if (depth_z_mapping_idc == 2)
for (i=0; i<256; i++)
value_z_i()
else if (depth_z_mapping_idc < 2) {
value_z_near()
value_z_far()
}
// intrinsic parameters
value_focal_length_x()
value_focal_length_y()
value_principal_point_x()
value_principal_point_y()
// extrinsic parameters
value_rotation_xy()
value_rotation_xz()
value_rotaion_yz()
horizontal_alligned_flag
if (!horizontal_alligned_flag)
for (i = 0; i <= num_views_minus1; i++)
{
value_rotation_xy_i()
for (i = 0; i <= num_views_minus1; i++) {
value_translation()
}
}

5 donde depth_z_mapping_idc especifica el procedimiento de cuantificación del mapa de profundidad utilizado para convertir el valor z del mundo real al valor d de profundidad de 8 bits. Cuando este valor es igual a 0, especifica que d es proporcional a 1/z, es decir $d/255 = (1/z - 1/z_{far}) / (1/z_{near} - 1/z_{far})$. Cuando este valor es igual a 1, especifica que d es proporcional a z, es decir $d/255 = (z - z_{near}) / (z_{far} - z_{near})$. Cuando este valor es igual a 2, especifica que la forma de asignación d a z se señala explícitamente. value_z_i() es la tabla de sintaxis para señalar el valor de z[i] para cada valor de profundidad de 8 bits igual d a i, cuando depth_z_mapping_idc es igual a 2. value_z_near() y value_z_far() son tablas de sintaxis para señalar los valores z_near y z_far respectivamente. value_focal_length_x() y value_focal_length_y() son las tablas de sintaxis para señalar los valores focal_length_x y focal_length_y, respectivamente, como parte del parámetro de cámara intrínsecos que pueden compartir todas las cámaras. value_principal_point_x() y value_principal_point_y() son las tablas de sintaxis para señalar los valores principal_point_x y principal_point_y, respectivamente, como parte del parámetro de cámara intrínseco que pueden compartir todas las cámaras.

15

[0113] La matriz de rotación R del parámetro de cámara extrínseco para una cámara se puede representar de la siguiente manera:

$$R = \begin{bmatrix} R_{yz} & 0 & 0 \\ 0 & R_{xz} & 0 \\ 0 & 0 & R_{xy} \end{bmatrix}$$

5 value_rotation_xy(), value_rotation_xz(), y value_rotation_yz() son las tablas de sintaxis para señalar el valor de los elementos diagonales de la matriz de rotación R . horizontal_aligned_flag igual a 1 indica que todas las cámaras están alineadas horizontalmente. value_rotation_xy_i() es la tabla de sintaxis para indicar el valor de la R_{xy} de la matriz de rotación R para la respuesta de la cámara a la visualización i . value_translation() es la tabla de sintaxis para indicar el valor de la traslación, como parte del parámetro de cámara extrínseco de la cámara correspondiente a la visualización i . Típicamente, value_syntax() se puede señalar como un valor de punto flotante.

10 **[0114]** En un ejemplo, las cámaras pueden estar alineadas horizontalmente y diferentes cámaras pueden corresponder a diferentes rangos de profundidad. A continuación, se muestra un diseño específico correspondiente a este ejemplo, donde los parámetros de la cámara pueden codificarse de manera más eficiente mediante el uso de codificación diferencial entre múltiples visualizaciones, por ejemplo:

seq_parameter_set_depth_extension() {
//intrinsic parameters
focal_length_precision
focal_length_x_l
focal_length_y_l_diff_x
principal_precision
principal_point_x_l
principal_point_y_l_diff_x
//extrinsic parameters
rotation_xy_half_pi
rotation_xz_half_pi
rotation_yz_half_pi
translation_precision
anchor_view_id_
zero_translation_present_flag
if (!zero_translation_present_flag)
translation_anchor_view_l
//depth range
z_near_precision
z_far_precision
z_near_integer

<code>z_far_integer</code>
<code>for (i = 0; i <= num_views_minus1; i++)</code>
<code> if (i!= anchor_view_id) {</code>
<code> translation_diff_anchor_view_l[i]</code>
<code> z_near_diff_anchor_view_l[i]</code>
<code> z_far_diff_anchor_view_l[i]</code>
<code> }</code>
<code>}</code>

[0115] En esta tabla de sintaxis, un valor de punto flotante V puede representarse como un valor decimal, con su precisión P , que representa el número de dígitos antes o después de cero, y un valor entero I , de modo que: $V=I \cdot 10^P$. Cuando I es un valor positivo, V también es un valor positivo, cuando I es un valor negativo, V también es un valor negativo. Los siguientes son los parámetros intrínsecos para las visualizaciones relacionadas con el SPS. A menos que se especifique o actualice, los parámetros son los mismos para cualquiera de las visualizaciones relacionadas:

- focal_length_precision especifica la precisión de los valores de focal_length_x y focal_length_y.
- focal_length_x_l especifica la parte entera del valor de focal_length_x.
 $focal_length_x = focal_length_x_l \cdot 10^{focal_length_precision}$
- focal_length_y_l_diff_x más focal_length_x_l especifica la parte entera del valor de focal_length_y.
 $focal_length_y = (focal_length_x_l + focal_length_y_l_diff_x) \cdot 10^{focal_length_precision}$
- principal_precision especifica la precisión de los valores de principal_point_x y principal_point_y.
- principal_point_x_l especifica la parte entera del valor de principal_point_x. $principal_point_x = principal_point_x_l \cdot 10^{principal_precision}$
- principal_point_y_l_diff_x plus principal_point_x especifica la parte entera del valor de principal_point_y.
 $principal_point_y = (principal_point_x_l + principal_point_y_l_diff_x) \cdot 10^{principal_precision}$

[0116] Los siguientes son los parámetros extrínsecos para las visualizaciones relacionadas con el SPS. A menos que se especifique o actualice, los parámetros son los mismos para cualquiera de las visualizaciones relacionadas.

- La matriz de rotación R para cada cámara se representa de la siguiente manera:

$$R = \begin{bmatrix} R_{yz} & 0 & 0 \\ 0 & R_{xz} & 0 \\ 0 & 0 & R_{xy} \end{bmatrix}$$

- rotation_kl_half_pi indica los elementos diagonales de la matriz de rotación R , con kl igual a xy , yz o xz , en el que $R_{kl} = (-1)^{rotation_kl_half_pi}$. Este indicador igual a 0 especifica $R_{kl}=1$; este indicador igual a 1 especifica $R_{kl}=-1$.
- translation_precision especifica la precisión de los valores de las traslaciones de todas las visualizaciones relacionadas. La precisión de los valores de traslación como se especifica en este SPS se aplica a todos los valores de traslación de los componentes de visualización que hacen referencia a este SPS.
- anchor_view_id especifica el view_id de la visualización, cuya traslación se usa como anclaje para calcular la traslación de las otras visualizaciones. La traslación de la visualización con view_id igual a anchor_view_id es igual a 0 cuando zero_translation_present_flag es igual a 0, de lo contrario, se señala la traslación.
- z_near_precision especifica la precisión del valor de z_near. La precisión de z_near como se especifica en este SPS se aplica a todos los valores z_near de los componentes de visualización que se refieren a este SPS.
- z_far_precision especifica la precisión del valor de z_near. La precisión de z_far como se especifica en este SPS se aplica a todos los valores z_far de los componentes de visualización que hacen referencia a este SPS.

- $z_near_integer$ especifica la parte entera del valor de z_near . $z_near = z_near_integer * 10^{z_near_precision}$

- $z_far_integer$ especifica la parte entera del valor de z_far . $z_far = z_far_integer * 10^{z_far_precision}$

5 (Cuando los valores del rango de profundidad son diferentes para diferentes visualizaciones, z_near y z_far especifican el rango de profundidad de la visualización de anclaje)

10 - $zero_translation_present_flag$ igual a 1 indica que la traslación de la visualización con $view_id$ igual a $anchor_view_id$ es 0; este valor igual a 0 indica que se señala la traslación de la visualización con $view_id$ igual a $anchor_view_id$.

- $translation_anchor_view_l$ especifica la parte entera de la traslación de la visualización de anclaje.

15 - $translation_anchor_view = translation_anchor_view_l * 10^{translation_precision}$

- $translation_anchor_view_l$ se deduce que es 0 cuando $zero_translation_present_flag$ es igual a 1.

20 - $translation_diff_anchor_view_l[i]$ más $translation_anchor_view_l$ especifica la parte entera de la traslación de la visualización con $view_id_equal$ en i , denotada como $translation_view[i]$.

- $translation_view[i] = (translation_anchor_view_l + translation_diff_anchor_view_l[i]) * 10^{translation_precision}$

25 (Denota la parte entera de la $translation_view[i]$ como $translation_view_l[i]$)

- $z_near_diff_anchor_view_l$ más $z_near_Integer$ especifica la parte entera del valor de profundidad más cercano de la visualización con id de visualización igual a i , denotado como $z_near[i]$.

30 - $z_near[i] = (z_near_diff_anchor_view_l[i] + z_near_Integer) * 10^{z_near_precision}$

(Denote la parte entera de z_cerca de $[i]$ como $z_near_l[i]$)

- $z_far_diff_anchor_view_l$ más $z_far_Integer$ especifica la parte entera del valor de profundidad más lejano de la visualización con $view_id$ igual a i , denotado como $z_far[i]$.

35 - $z_far[i] = (z_far_diff_anchor_view_l[i] + z_far_Integer) * 10^{z_far_precision}$

(Denote la parte entera de z_far de $[i]$ como $z_far_l[i]$)

40 **[0117]** En este diseño de ejemplo, un valor se representa como $V = l * 10^P$. En otro ejemplo, el valor puede representarse con otras bases, como $V = l * b^P$ en el que b puede ser igual a 2, 8 o 16. En un ejemplo, $focal_length_x_l$, $principal_point_x_l$, $translation_anchor_view_l$, $z_near_integer$ y $z_far_integer$ pueden señalarse como codificación de longitud fija, por ejemplo, con una longitud por defecto de 32 bits, o con una longitud señalada de N bits.

45 **[0118]** En algunos ejemplos, el rango de profundidad o el parámetro de cámara extrínseco (por ejemplo, traslación) de una visualización puede cambiar en función de la imagen o puede actualizarse. El rango de profundidad actualizado o los parámetros de la cámara pueden aplicarse a los componentes de visualización de la unidad de acceso actual y los siguientes componentes de visualización en el flujo de bits, hasta que un nuevo PPS contenga la extensión de profundidad PPS después de que el PPS actual actualice esos valores para las visualizaciones relacionadas. En otro ejemplo, un PPS que contiene una extensión de profundidad PPS puede asignarse como un tipo de unidad NAL diferente, en ese caso, la propia extensión de profundidad PPS forma un RBSP.

50 **[0119]** En otro ejemplo, un conjunto de parámetros de visualización con un tipo de unidad NAL diferente puede introducirse para señalar el cambio del rango de profundidad y la traslación. El rango de profundidad y la traslación de la cámara pueden cambiar en función de la imagen. El rango de profundidad o los parámetros de la cámara pueden actualizados ser aplicables a los componentes de visualización de la unidad de acceso actual y los siguientes componentes de visualización en el flujo de bits, hasta que un nuevo VPS después del VPS actual actualice esos valores para las visualizaciones relacionadas. Los elementos sintácticos pueden tener la misma semántica que para la extensión PPS. Un conjunto de parámetros de visualización puede referirse directamente al SPS actualmente activo que contiene la extensión de profundidad.

60 **[0120]** En algunos ejemplos, el rango de profundidad y la traslación pueden cambiar drásticamente después de un tiempo, por lo que puede ser menos eficiente si el VPS solo se refiere directamente a SPS. Una forma de resolver este posible problema puede ser calcular un conjunto de parámetros de visualización después de analizar la cabecera de fragmento del primer componente de visualización de profundidad para saber a qué PPS se refiere el VPS, cuando el

65

conjunto de parámetros de visualización se refiere a un PPS con la extensión de profundidad. En otro ejemplo, una identificación de PPS se puede poner en el VPS para resolver el problema.

5 **[0121]** En algunos ejemplos, algunos conjuntos de parámetros de visualización pueden recibir la misma prioridad que SPS y PPS, por lo cual pueden señalarse fuera de banda. En este ejemplo, se puede introducir un indicador para que el VPS actualice los valores en el SPS y el siguiente VPS normal se refiere solo a los valores actualizados. En este ejemplo, la trama al que se puede señalar el VPS.

10 **[0122]** El rango de profundidad y la traslación de la cámara pueden cambiar en función de la imagen. El rango de profundidad actualizado o los parámetros de la cámara pueden aplicarse a los componentes de visualización de la unidad de acceso actual y los siguientes componentes de visualización en el flujo de bits, hasta que un nuevo PPS contenga la extensión de profundidad PPS después de que el PPS actual actualice esos valores para las visualizaciones relacionadas.

15 **[0123]** En un ejemplo, la sintaxis del conjunto de parámetros de imagen puede ser la siguiente:

```

pic_parameter_set_rbsp( ) {
    pic_parameter set_id
    ...
    if (profile_idc == 3DV_profile) {
        update_depth_translation_flag
        if (update_depth_translation_flag)
            pic_parameter_set_depth_ext
        ( )
    }
    rbsp_trailing_bits( )
}
    
```

```

pic_parameter_set_depth_ext( ) {
    base_pic_parameter_set_id_
    z_near_diff_integer_anchor
    z_far_diff_integer_anchor
    if (!zero_translation_present_flag)
        delta_translation_anchor_view_l
    for (i = 0; i <= num_views_minus1; i++)
        if (i != anchor_view_id) {
            delta_translation_diff_anchor_view_l[i ]
            z_near_diff_update_anchor_view_l[i ]
            z_far_diff_update_anchor_view_l[i ]
        }
    rbsp_trailing_bits( )
}
    
```

[0124] Donde

- 5 - base_pic_parameter_set_id especifica el ID del conjunto de parámetros de imagen al que se refiere el conjunto de parámetros de la imagen actual. El conjunto de parámetros correspondiente puede ser un conjunto de parámetros de imagen normal o una extensión de conjunto de parámetros de imagen.
- 10 - z_near_diff_integer_anchor más z_near_integer especifica la parte entera de la nueva profundidad más cercana de la visualización de anclaje.
- 15 - z_far_diff_integer_anchor más z_far_integer especifica la parte entera de la nueva profundidad más lejana de la visualización de anclaje.
- 20 - delta_translation_anchor_view_l plus especifica la diferencia entre las partes enteras, la nueva traslación de la visualización de anclaje y la traslación previa de la visualización de anclaje. Se infiere que este valor es 0 si zero_translation_present_flag es igual a 1.
- 25 - delta_translation_diff_anchor_view_l[i] más translation_diff_anchor_view_l[i], como se especifica en el SPS, especifica la diferencia entre translation_anchor_view_l y la parte entera de la nueva traslación de la visualización con view_id igual a i.
- 30 -
$$\text{new_translation_view_l}[i] = (\text{delta_translation_diff_anchor_view_l}[i] + \text{translation_diff_anchor_view_l}[i] + \text{new_translation_anchor_view_l}) * 10^{\text{translation_precision}}$$
- o
- $$\text{new_translation_view_l}[i] = (\text{delta_translation_diff_anchor_view_l}[i] + \text{translation_view_l}[i] + \text{new_translation_anchor_view_l}) * 10^{\text{translation_precision}}$$

[0125] En algunos ejemplos, las traslaciones de la visualización de anclaje son siempre 0, por lo que la segunda fórmula puede ser más simple. En otros ejemplos, este valor se puede usar directamente para señalar la diferencia entre las partes enteras de la nueva traslación de la visualización con ID de visualización igual ai y la traslación correspondiente señalada en SPS.

- 35 - translation_diff_update_anchor_view_l[i] especifica la diferencia entre las partes enteras la nueva traslación de la visualización con view_id igual a i y la traslación calculada con la señalización en el SPS para la misma visualización.
- 40 -
$$\text{new_translation_view_l}[i] = (\text{translation_diff_update_anchor_view_l}[i] + \text{translation_view_l}[i]) * 10^{\text{translation_precision}}$$

[0126] En este ejemplo, el nombre de la sintaxis puede cambiarse de

45 delta_translation_diff_anchor_view_l[i] a translation_diff_update_anchor_view_l[i]. Los valores previos a los que se refiere el PPS son los valores definidos en el SPS, si el base_pic_parameter_set_id corresponde a un conjunto de parámetros de imagen normal; de lo contrario (el base_pic_parmeter_set_id corresponde a una extensión de profundidad de conjunto de parámetros de imagen) los valores previos son los valores señalados por el conjunto de parámetros de imagen de referencia (base).

50 [0127] En otro, un PPS que contiene una extensión de profundidad PPS puede asignarse como un tipo de unidad NAL diferente; en ese caso, la propia extensión de profundidad PPS forma un RBSP:

pic_parameter_set_depth_ext_rbsp () {
base_pic_parameter_set_id
pic_parameter_set_id
z_near_diff_integer_anchor
...
rbsp_trailing_bits()
}

[0128] Donde

- 5 - base_pic_parameter_set_id especifica el id del parámetro de imagen establecido al que se refiere el conjunto de parámetros de la imagen actual. El conjunto de parámetros correspondiente puede ser un conjunto de parámetros de imagen normal o una extensión de conjunto de parámetros de imagen.
- 10 - pic_parameter_set_id especifica el ID de conjunto de parámetros de imagen de la extensión de conjunto de parámetros de imagen actual.

[0129] En un ejemplo, los conjuntos de parámetros de imagen normales y las extensiones de profundidad de conjunto de parámetros de imagen comparten el mismo sistema de numeración.

15 [0130] En algunos ejemplos, se puede introducir un conjunto de parámetros de visualización (por ejemplo, con un tipo de unidad NAL diferente) para señalar los cambios del rango de profundidad y la traslación.

view_parameter_set_rbsp() {
z_near_diff_integer_anchor
z_far_diff_integer_anchor
if (!zero_translation_present_flag)
delta_translation_anchor_view_l
for (i = 0; i <= num_views_minus1; i++)
if (i != anchor_view_id) {
delta_translation_diff_anchor_view_l[i]
z_near_diff_update_anchor_view_l[i]
z_far_diff_update_anchor_view_l[i]
}
rbsp_trailing_bits()
}

20 En algunos ejemplos, el rango de profundidad y la traslación de la cámara pueden cambiar en función de la imagen. El rango de profundidad o los parámetros de la cámara pueden actualizados ser aplicables a los componentes de visualización de la unidad de acceso actual y los siguientes componentes de visualización en el flujo de bits, hasta que un nuevo VPS después del VPS actual actualice esos valores para las visualizaciones relacionadas. Si hay un VPS presente en una unidad de acceso de flujo de bits, el VPS puede colocarse antes de cualquier componente de visualización de una unidad de acceso. Los elementos sintácticos pueden tener la misma semántica que para la extensión PPS. Un conjunto de parámetros de visualización puede referirse directamente al SPS actualmente activo que contiene la extensión de profundidad.

25

5 **[0131]** En algunos ejemplos, el rango de profundidad y la traslación pueden cambiar drásticamente después de un tiempo, por lo que puede no ser tan eficiente si el VPS solo se refiere directamente a SPS. En este ejemplo, un conjunto de parámetros de visualización puede referirse a un PPS con la extensión de profundidad, y se puede calcular después de analizar la cabecera de fragmento del primer componente de visualización de profundidad para saber a qué PPS se refiere el VPS. En otro ejemplo, una ID de PPS puede colocarse en el VPS, como se muestra a continuación:

```

view_parameter_set_rbsp() {
    pic_parameter_set_id
    z_near_integer_anchor
    ...
    rbsp_trailing_bits()
}
    
```

10 **[0132]** En algunos ejemplos, algunos conjuntos de parámetros de visualización pueden recibir la misma prioridad que SPS y PPS y, por lo tanto, pueden señalarse fuera de banda. En este ejemplo, se puede introducir un indicador para que el VPS actualice los valores en el SPS y el siguiente VPS normal pueda hacer referencia a los valores actualizados. En este ejemplo, se señala la primera trama a la que se aplica el VPS. Si tal VPS no se recibe descodificando una trama de referencia con un núm. de trama igual a start_frame_num, se puede considerar como
 15 perdido. En otro ejemplo, también se puede señalar un valor de POC.

```

view_parameter_set_rbsp() {
    update_SPS_parameter_flag
    if (update_SPS_parameter_flag)
        start_frame_num
    z_near_integer_anchor
    ...
    rbsp_trailing_bits()
}
    
```

20 **[0133]** La FIG. 4A es un diagrama conceptual que ilustra un patrón de predicción de MVC de ejemplo. La FIG. 4A proporciona un esquema de codificación de ejemplo para la predicción entre visualizaciones. En general, una trama codificada de datos de vídeo de MVC puede codificarse de forma predictiva espacialmente, temporalmente y/o con referencia a tramas de otras visualizaciones en una ubicación temporal común. Por consiguiente, las visualizaciones de referencia, a partir de las cuales se predicen otras visualizaciones, se descodifican en general antes de las visualizaciones para las que las visualizaciones de referencia actúan como referencia, de modo que estas
 25 visualizaciones descodificadas se puedan usar como referencia cuando se descodifican visualizaciones referenciales. El orden de descodificación no necesariamente corresponde al orden de los view_ids. Por lo tanto, el orden de descodificación de visualizaciones se describe utilizando índices de orden de visualización. Los índices de orden de visualización son índices que indican el orden de descodificación de los correspondientes componentes de visualización en una unidad de acceso.

30 **[0134]** En el ejemplo de la FIG. 4A, se muestran ocho visualizaciones (que tienen identificadores de visualización "S0" a "S7"), y se ilustran doce ubicaciones temporales ("T0" a "T11") para cada visualización. Es decir, cada fila en la FIG. 4A corresponde a una visualización, mientras que cada columna indica una ubicación temporal. Aunque la MVC tiene una denominada visualización de base que es descodificable mediante los descodificadores de H.264/AVC y los pares de visualizaciones estéreo podrían soportarse también mediante MVC, la ventaja de la MVC es que podría
 35 soportar un ejemplo que usa más de dos visualizaciones como una entrada de vídeo 3D y descodifica este vídeo 3D representado por las múltiples visualizaciones. Un renderizador de un cliente que tiene un descodificador de MVC puede esperar contenido de vídeo tridimensional con múltiples visualizaciones.

[0135] Las tramas de la FIG. 4A están indicadas en la intersección de cada fila y cada columna en la FIG. 4A usando un bloque sombreado que incluye una letra, designando si la trama correspondiente está intracodificada (es decir, una trama I) o intercodificada en una dirección (es decir, como una trama P) o en múltiples direcciones (es decir, como una trama B). En general, las predicciones se indican mediante flechas, donde la trama a la que se apunta utiliza el objeto desde el que se apunta como referencia de predicción. Por ejemplo, la trama P de la visualización S2 en la ubicación temporal T0 se predice a partir de la trama I de la visualización S0 en la ubicación temporal T0.

[0136] Al igual que con la codificación de vídeo de visualización única, las tramas de una secuencia de vídeo de codificación de vídeo multi-visualización pueden codificarse predictivamente con respecto a las tramas en diferentes ubicaciones temporales. Por ejemplo, la trama B de la visualización S0 en la ubicación temporal T1 tiene una flecha apuntando a la misma desde la trama I de la visualización S0 en la ubicación temporal T0, lo cual indica que la trama B se predice a partir de la trama I. Adicionalmente, sin embargo, en el contexto de la codificación de vídeo multi-visualización, las tramas pueden predecirse entre visualizaciones. Es decir, un componente de visualización puede utilizar los componentes de visualización en otras visualizaciones como referencia. En la MVC, por ejemplo, la predicción entre visualizaciones se realiza como si el componente de visualización en otra visualización es una referencia de predicción inter. Las posibles referencias entre visualizaciones se señalan en la extensión MVC del conjunto de parámetros de secuencia (SPS) y pueden modificarse mediante el proceso de construcción de la lista de imágenes de referencia, que permite un ordenamiento flexible de las referencias de predicción entre predicciones o entre visualizaciones. La predicción entre visualizaciones en MVC puede soportarse mediante la compensación de disparidad, que usa la sintaxis de la compensación de movimiento H.264/AVC, pero permite que una imagen en una visualización diferente se ponga como una imagen de referencia.

[0137] La FIG. 4A proporciona varios ejemplos de predicción entre visualizaciones. Las tramas de visualización S1, en el ejemplo de la FIG. 4A, se ilustran como predichas a partir de tramas en diferentes ubicaciones temporales de la visualización S1, así como también como entre visualizaciones predichas a partir de tramas de tramas de visualizaciones S0 y S2 en las mismas ubicaciones temporales. Por ejemplo, la trama-B de la visualización S1 en la ubicación temporal T1 se predice a partir de cada una de las tramas-B de la visualización S1 en las ubicaciones temporales T0 y T2, así como las tramas-B de las visualizaciones S0 y S2 en la ubicación temporal T1.

[0138] En el ejemplo de la FIG. 4A, mayúscula "B" y minúscula "b" están destinadas a indicar diferentes relaciones jerárquicas entre tramas, en lugar de diferentes metodologías de codificación. En general, las tramas con "B" mayúscula están relativamente más altas en la jerarquía de predicción que las tramas con "b" minúscula. La FIG. 4A también ilustra las variaciones en la jerarquía de predicción utilizando diferentes niveles de sombreado, donde las tramas con una mayor magnitud de sombreado (es decir, relativamente más oscuras) están más altas en la jerarquía de predicción que aquellas tramas que tienen menos sombreado (es decir, relativamente más claras). Por ejemplo, todas las tramas I en la FIG. 4A se ilustran con sombreado completo, mientras que las tramas P tienen un sombreado algo más claro, y las tramas B (y las tramas b minúscula) tienen varios niveles de sombreado entre sí, pero siempre más claro que el sombreado de las tramas P y las tramas I.

[0139] La FIG. 4B es un diagrama conceptual que ilustra la codificación de vídeo de múltiples visualizaciones (MVC) compatible con la codificación de vídeo de múltiples visualizaciones más profundidad (MVD) para dos visualizaciones. Cada "visualización" en un flujo de bits codificado MVC puede incluir una pluralidad de componentes de visualización, correspondiendo cada uno a instancias temporales específicas durante la reproducción. Cada componente de visualización puede incluir datos de textura y datos de profundidad. La FIG. 4B es una instancia específica de MVC que emplea codificación de vídeo estéreo, ya que solo hay dos visualizaciones. Sin embargo, los datos de un flujo de bits ensamblado de acuerdo con la FIG. 4B se puede usar para producir más de dos visualizaciones de salida, debido a la información de profundidad incluida, que puede utilizarse para sintetizar otras visualizaciones.

[0140] En un ejemplo, se pueden utilizar una o más cámaras para capturar y proporcionar datos de vídeo que se usan para producir los componentes de textura y profundidad. Los datos de vídeo capturados por la cámara es información de textura. En un ejemplo, los pares de cámara estéreo se pueden usar para calcular el mapa de profundidad para el componente de textura. En otro ejemplo, un sensor de profundidad (por ejemplo, RADAR, LADAR o similar) puede determinar mapas de profundidad para una visualización. La determinación de un componente de profundidad puede realizarse mediante una unidad de procesamiento de profundidad 21 del dispositivo de origen 12 de la FIG. 1. En un ejemplo, la unidad de procesamiento de profundidad puede incorporarse en el descodificador de vídeo 22.

[0141] En el ejemplo de la FIG. 2B, en cada instante en el tiempo, se pueden obtener dos visualizaciones, visualización 0 y visualización 1. La visualización 0 comprende el componente de textura 90A (por ejemplo, textura 0) y el componente de profundidad 90B (por ejemplo, profundidad 0), y la visualización 1 comprende el componente de textura 92A (por ejemplo, textura 1) y el componente de profundidad 92B (por ejemplo, profundidad 1). En el ejemplo de la FIG. 4B, cada uno de los componentes 90A, 90B, 92A y 92B puede referirse a los componentes correspondientes de las dos visualizaciones en una instancia temporal común, f0, por ejemplo. Los mismos componentes se pueden determinar en instancias temporales posteriores, f1, f2, etc. Si bien las técnicas se describen con referencia a los componentes en f0, debe entenderse que las mismas técnicas pueden aplicarse de manera similar a los componentes en otras instancias temporales, f1, f2, etc.

[0142] La textura de las dos visualizaciones puede codificarse en una codificación MVD compatible con MVC, que en general puede parecerse a MVC extendida de acuerdo con las técnicas de esta divulgación para adaptarse incluyendo información de profundidad en el mismo flujo de bits que la información de textura, mientras se mantiene su separación.

[0143] Usando las técnicas de esta divulgación, los mapas de profundidad de cada una de las dos visualizaciones pueden predecirse usando otra información correspondiente a la misma visualización o a la otra visualización. En un ejemplo, el mapa de profundidad asociado con un componente de profundidad puede predecirse utilizando la predicción de movimiento desde mapa de profundidad a textura. Por ejemplo, el componente de profundidad 90B puede predecirse usando el componente de textura 90A, y el componente de profundidad 92B puede predecirse usando el componente de textura 92A.

[0144] En otro ejemplo, el mapa de profundidad de una visualización puede predecirse utilizando la predicción de disparidad entre visualizaciones de una visualización a la otra (la predicción entre visualizaciones en MVC). Por ejemplo, el componente de profundidad 92B puede predecirse con relación al componente de profundidad 90B. De la misma manera, el componente de textura 92A se puede predecir con relación al componente de textura 90A.

[0145] En otro ejemplo más, el mapa de profundidad de una visualización puede predecirse usando la predicción de síntesis de visualización, que puede utilizar parámetros de cámara asociados con la visualización. Si los parámetros de cámara de cada una de las visualizaciones están disponibles, se puede sintetizar una visualización actual virtual a partir de una visualización de referencia y su correspondiente mapa de profundidad usando técnicas tales como, por ejemplo, técnicas de DIBR. La visualización virtual sintetizada se puede utilizar para predecir y codificar otra visualización. En un ejemplo, para la predicción de síntesis de visualización de mapa de profundidad, la visualización de referencia puede ser el propio mapa de profundidad. Por lo tanto, los parámetros de la cámara y un mapa de profundidad de referencia pueden ser suficientes para sintetizar un mapa de profundidad actual virtual.

[0146] Como se analizó anteriormente, las técnicas de esta divulgación pueden incluir parámetros de la cámara de señalización para una visualización. Los parámetros de cámara pueden incluir, por ejemplo, parámetros intrínsecos y parámetros extrínsecos. Los parámetros intrínsecos pueden incluir, por ejemplo, la distancia focal y el desplazamiento del punto principal en la dirección horizontal. Los parámetros extrínsecos pueden incluir, por ejemplo, la ubicación de la cámara horizontal del mundo real para cada visualización. Típicamente, cada visualización de una secuencia de múltiples visualizaciones compartirá los mismos parámetros intrínsecos. Por lo tanto, estos parámetros pueden señalarse una vez para la secuencia, de modo que los parámetros intrínsecos señalados se apliquen a todas las visualizaciones de la secuencia.

[0147] En un ejemplo, los parámetros de cámara intrínsecos y extrínsecos pueden señalarse en un conjunto de parámetros de secuencia (SPS) de datos tridimensionales de vídeo (3DV) o extensión 3DV SPS. La extensión SPS 3DV puede describir aún más el valor de profundidad a la asignación de valores z del mundo real, lo cual puede incluir funciones de conversión de rango de profundidad y profundidad por defecto a valor z, y/o una tabla que contiene el valor de profundidad al valor Z real para cada configuración de cámara. La extensión SPS 3DV también puede describir una relación de predicción de visualizaciones de mapas de profundidad y visualizaciones de textura, que pueden utilizarse en la predicción de movimiento desde el mapa de textura al de profundidad.

[0148] La FIG. 4C es un diagrama conceptual que ilustra la codificación MVD compatible con MVC para tres o más visualizaciones. Para fines ilustrativos, el ejemplo de la FIG. 4C muestra 3 visualizaciones, pero se debe entender que estas técnicas se pueden utilizar para 3 o más visualizaciones. En un ejemplo, cada una de las visualizaciones puede obtenerse mediante una cámara desde una perspectiva diferente y cada visualización puede comprender información de textura y profundidad. En otros ejemplos, se pueden obtener algunas visualizaciones desde diferentes perspectivas, y se pueden sintetizar otras visualizaciones usando los componentes de las visualizaciones obtenidas. En cada instante en el tiempo, se pueden obtener tres visualizaciones, visualización 0, visualización 1 y visualización 2. La visualización 0 comprende el componente de textura 94A (por ejemplo, la textura 0) y el componente de profundidad 94B (por ejemplo, profundidad 0), la visualización 1 comprende el componente de textura 96A (por ejemplo, la textura 1) y el componente de profundidad 96B (por ejemplo, profundidad 1) y la visualización 2 comprende el componente de textura 98A (por ejemplo, la textura 2) y el componente de profundidad 98B (por ejemplo, profundidad 2).

[0149] Los mapas de profundidad asociados con cada una de las visualizaciones pueden predecirse, como se analizó anteriormente, utilizando mapas de profundidad de otra visualización o usando información de textura de la misma visualización. Además, la información de textura de una visualización puede predecirse usando información de textura de una o más visualizaciones diferentes. En el ejemplo de la FIG. 4B, para una tercera visualización, la textura también puede predecirse a partir de los mapas de textura y profundidad de una o más visualizaciones diferentes. En este ejemplo, la tercera visualización puede predecirse usando los modos de predicción entre visualizaciones definidos en MVC. Los bloques de la tercera visualización también pueden predecirse utilizando la predicción de síntesis de visualización, que puede basarse en los mapas de textura y profundidad de la primera visualización y la segunda visualización.

[0150] Como se indicó anteriormente, las técnicas de esta divulgación pueden utilizar la extensión SPS 3DV para señalar ciertos parámetros que pueden usarse para predecir información de textura y profundidad. La extensión SPS 3DV puede describir los parámetros de la cámara, el valor de profundidad a la asignación de valores Z del mundo real y/o la relación de predicción de las visualizaciones del mapa de profundidad y las visualizaciones de textura.

[0151] Los parámetros de la cámara pueden incluir parámetros intrínsecos y parámetros extrínsecos. Los parámetros intrínsecos pueden incluir la distancia focal y el desplazamiento del punto principal en la dirección horizontal. Las secuencias de múltiples visualizaciones en general comparten la misma distancia focal y el mismo punto de desplazamiento principal en la dirección horizontal. Por lo tanto, en la implementación, para parámetros intrínsecos, se pueden utilizar dos números flotantes para todas las visualizaciones. Los parámetros extrínsecos pueden incluir la ubicación horizontal de la cámara de cada visualización. Las secuencias de múltiples visualizaciones en general son capturadas por una matriz de cámaras paralelas donde todas las cámaras pueden tener la misma orientación en las coordenadas mundiales. Las cámaras pueden colocarse en una línea para que sus posiciones solo difieran en la dirección horizontal. Por lo tanto, la cantidad de parámetros extrínsecos es igual a la cantidad de visualizaciones. Por ejemplo, para el caso de N visualizaciones, se pueden utilizar N números flotantes para parámetros extrínsecos.

[0152] En un ejemplo, la asignación de valores de profundidad a valores z del mundo real se puede lograr utilizando funciones de conversión de rango de profundidad y profundidad por defecto a valor z . En otros ejemplos, donde otras funciones de conversión pueden diferir de las funciones por defecto, se puede incluir una tabla que contiene el valor de profundidad al valor z del mundo real para cada configuración de cámara. En un ejemplo, la conversión entre el valor de profundidad y el valor de z del mundo real se puede usar en la síntesis de visualización para predecir visualizaciones, por ejemplo, visualización virtual, desde una o más visualizaciones codificadas. Es decir, un valor de profundidad asignado a un valor z del mundo real se puede expresar en un rango de profundidad del mundo real, lo cual puede indicar un desplazamiento horizontal necesario al producir la visualización virtual para proporcionar el efecto 3D adecuado.

[0153] La relación de predicción de las visualizaciones del mapa de profundidad y las visualizaciones de textura también puede describirse mediante una extensión SPS 3DV. La relación de predicción de las visualizaciones del mapa de profundidad sigue la relación de predicción de la síntesis de visualización. Por lo tanto, si la textura de la visualización 0 depende de la textura de la visualización 1 basada en la predicción entre visualizaciones, el mapa de profundidad de la visualización 0 puede depender de la visualización 1. Si la textura de la visualización 0 no depende de la textura de la visualización 1, entonces el mapa de profundidad de la visualización 0 probablemente no dependa del mapa de profundidad de la visualización 1. Para cada una de las relaciones de predicción anteriores, es decir, si el mapa de profundidad de la visualización 0 depende del mapa de profundidad de la visualización 1 bajo la condición de que la textura de la visualización 0 dependa de la textura de la visualización 1, se puede señalar un indicador para indicar si existe una relación predictiva. Además, se puede agregar un mensaje SEI para proporcionar más detalles sobre los cambios en la relación de predicción. De esa manera, un mensaje SEI puede estar asociado a una subsecuencia que tiene el indicador específico igual a verdadero mientras que otro mensaje SEI asociado a otra subsecuencia que tiene el indicador específico igual a falso.

[0154] Como se analizó anteriormente, las unidades VCL NAL correspondientes a la misma instancia temporal para una visualización se pueden tratar como un componente de visualización común. Las técnicas de esta divulgación pueden proporcionar datos de textura de codificación y datos de mapas de profundidad para una visualización consecutivamente en el flujo de bits. Típicamente, un componente de profundidad, si existe, siempre sigue su componente de textura correspondiente en el orden del flujo de bits. Para separar la información de mapas de profundidad y textura de un componente de visualización, estas técnicas de esta divulgación proponen agregar un delimitador en el flujo de bits, por ejemplo, un delimitador de unidad NAL que indica el final de los datos de textura y el comienzo de la información de profundidad para el componente de visualización.

[0155] Además o de forma alternativa, las unidades VCL NAL de textura y profundidad pueden incluir un indicador en la cabecera de la unidad NAL que indica si la unidad NAL incluye información de textura o profundidad. Este indicador puede denominarse indicador de profundidad, teniendo un valor de 0 para indicar que la unidad NAL incluye información de textura, o un valor de 1 para indicar que la unidad NAL incluye información de profundidad. Esta cabecera de unidad NAL puede corresponder a un nuevo tipo de unidad NAL. Además, se puede proporcionar información de cabecera de la unidad NAL para indicar si la unidad NAL se usa para predecir cualquier imagen de textura codificada de una visualización. Por ejemplo, se puede agregar un bit en la unidad VCL NAL de un mapa de profundidad, para indicar si esta unidad VCL NAL se usará para predecir cualquier imagen codificada de la textura de una visualización.

[0156] La FIG. 4D es un diagrama conceptual que ilustra la codificación MVD compatible con MVC para múltiples visualizaciones. En este ejemplo, pueden obtenerse datos de vídeo correspondientes a diferentes visualizaciones, por ejemplo, cada uno mediante una cámara desde una perspectiva diferente, en diferentes instancias temporales. En este ejemplo, las visualizaciones 200 y 220 pueden corresponder a dos visualizaciones diferentes. Los datos de vídeo asociados con los componentes de visualización 202A y 222A pueden obtenerse al mismo tiempo y los datos de vídeo asociados con los componentes de visualización 202B y 222B pueden obtenerse en el mismo instante de tiempo, que

es posterior al momento en que se obtuvieron los datos de vídeo asociados con los componentes de visualización 202A y 222A.

5 **[0157]** Cada uno de los componentes de visualización 202A, 222A, 202B y 222B puede comprender un componente de textura y un componente de profundidad. El componente de textura puede corresponder a datos capturados por una cámara. Los componentes de profundidad se pueden codificar en relación con los componentes de la misma visualización u otras visualizaciones. En el ejemplo de la FIG. 4D, el componente de profundidad 206A se predice con relación al componente de textura 204A del mismo componente de visualización 202A. En este ejemplo, el componente de profundidad 206B se predice con relación al componente de profundidad 206A de otro componente de visualización 202A de la misma visualización (por ejemplo, visualización 200), ilustrando así un ejemplo usando la predicción intra-visualización. En otro ejemplo, se puede usar la predicción entre visualizaciones, tal como predecir el componente de profundidad 226A de la visualización 220 con relación al componente de profundidad 206A de otra visualización 200.

15 **[0158]** En un ejemplo, la predicción de síntesis de visualización puede utilizarse para sintetizar una visualización virtual 230 usando la visualización capturada 200. En este ejemplo, el componente de visualización virtual 232B puede comprender el componente de textura 234B, que puede formarse, por ejemplo, interpolarse basándose en datos de uno o más componentes de otras visualizaciones, por ejemplo, el componente de textura 204B y el componente de profundidad 206B del componente de visualización 202B. Un componente virtual sintetizado también se puede utilizar para predecir otros componentes. En este ejemplo, el componente de profundidad 226B del componente de visualización 222B se predice con relación al componente de textura 234B.

20 **[0159]** La FIG. 5 es un diagrama de flujo que ilustra una operación de ejemplo de un codificador de vídeo coherente con esta divulgación. La técnica de la FIG. 5 se describirá desde la perspectiva del codificador de vídeo 22 de la FIG. 2, aunque otros dispositivos pueden realizar técnicas similares.

25 **[0160]** El codificador de vídeo 22 puede obtener una trama de datos de vídeo de textura (102). En otros ejemplos, el codificador de vídeo 22 puede recibir un fragmento u otra unidad codificada que incluye uno o más bloques de datos de vídeo. En algunos ejemplos, las tramas de dos o más visualizaciones pueden obtenerse mediante el codificador de vídeo 22, donde las dos o más visualizaciones pueden corresponder a dos o más cámaras. El codificador de vídeo 22 puede codificar datos de textura para un componente de visualización de una visualización (104), donde el componente de visualización incluye tanto la trama (o fragmento) de datos de vídeo de textura, como los datos de vídeo de profundidad de trama (o fragmento) para un instancia temporal común. La codificación puede incluir cualquiera o todas las codificaciones intra y entre visualizaciones de la trama.

30 **[0161]** El codificador de vídeo 22 puede recibir además un mapa de profundidad correspondiente al componente de visualización (106). El mapa de profundidad puede incluir información de profundidad para regiones de píxeles en la trama de vídeo. El mapa de profundidad puede tener una resolución igual o diferente a la de los componentes de textura. En un ejemplo, el mapa de profundidad puede determinarse mediante una unidad de procesamiento de profundidad.

35 **[0162]** El descodificador de vídeo 22 puede determinar la muestra de referencia para los bloques del mapa de profundidad (108). En un ejemplo, la muestra de referencia puede ser un componente en el mismo componente de visualización, un componente en la misma visualización pero un componente de visualización temporalmente independiente, o un componente en una visualización diferente. En un ejemplo, la determinación de las muestras de referencia puede incluir el análisis de datos de textura para los mismos componentes de visualización o la profundidad de un componente de visualización codificado previamente en la misma visualización. El descodificador de vídeo 22 puede entonces codificar el mapa de profundidad con relación a la muestra de referencia determinada (110). El descodificador de vídeo 22 puede proporcionar información de dependencia predictiva que indica la muestra de referencia determinada (112). La información puede incluir dependencias predictivas asociadas con la predicción del mapa de profundidad. La información de dependencia predictiva puede describir tramas de referencia o fragmentos de referencia usados para predecir datos de la trama o fragmento actual codificados (por ejemplo, un identificador de visualización, un valor de POC, o similar).

40 **[0163]** Los componentes codificados de cada componente de visualización pueden encapsularse juntos en un flujo de bits, mientras permanecen separados, incluyendo un indicador de profundidad en la cabecera de la unidad NAL para indicar si la unidad NAL corresponde a un componente de textura o un componente de profundidad. En un ejemplo, encapsular los datos de textura y profundidad puede incluir el uso de unidades NAL para cada componente de textura y profundidad, donde se puede utilizar un indicador en la cabecera de la unidad NAL para indicar si la unidad NAL es una unidad NAL de textura o profundidad. Además, se puede proporcionar una unidad NAL delimitadora entre los datos de textura y la unidad NAL de datos de profundidad en el flujo de bits común para indicar dónde termina la unidad NAL de textura y dónde comienza la unidad NAL de profundidad.

45 **[0164]** En un ejemplo, encapsular los datos de textura y profundidad puede comprender encapsular los datos de textura como una primera unidad de capa de abstracción de red (NAL) y encapsular los datos de profundidad como una segunda unidad NAL, donde la primera unidad NAL y la segunda unidad NAL corresponden a instancias del mismo tipo de unidad NAL. En un ejemplo, la primera unidad NAL puede comprender un primera cabecera de unidad NAL

que incluye un indicador indicando si la primera unidad NAL incluye información de textura o profundidad, y la segunda unidad NAL puede comprender un segunda cabecera de unidad NAL incluyendo un indicador indicando si la segunda unidad NAL incluye información de textura o profundidad. En un ejemplo, un valor del indicador de la primera cabecera de la unidad NAL puede configurarse para indicar que la primera unidad NAL incluye información de textura y un valor del indicador de la segunda cabecera de unidad NAL puede configurarse para indicar que la segunda unidad NAL incluye información de profundidad. De esta manera, durante el procesamiento del flujo de bits que contiene tanto la información de textura como la de profundidad, los indicadores en las cabeceras de las unidades NAL se pueden utilizar para determinar dónde finalizan los datos de textura y comienzan los datos de profundidad. Por lo tanto, separa fácilmente los dos componentes aunque los dos componentes estén encapsulados en un flujo de bits común. Esto también puede ser útil en casos donde el flujo de bits se puede transmitir a dispositivos que soportan pantallas 3D y dispositivos que soportan visualización en 2D, donde se puede utilizar un filtrado simple en la unidad NAL para proporcionar solo información de textura al dispositivo de visualización 2D.

[0165] Usando las técnicas de esta divulgación, la información que indica una relación predictiva entre los componentes de textura y profundidad puede señalarse con el flujo de bits. Las relaciones predictivas pueden indicar si un componente de profundidad puede predecirse usando la predicción intra o entre visualizaciones, y si un componente de visualización virtual puede formarse a partir de componentes de textura y/o componente de profundidad de una o más visualizaciones diferentes. Otra información señalada puede incluir parámetros de cámara asociados con cada una de las diferentes visualizaciones. Usando los parámetros de la cámara, la información de textura y profundidad para las diferentes visualizaciones puede codificarse en relación con la información de textura y profundidad desde una visualización de referencia. De esta manera, la información de textura y profundidad de un subconjunto de las visualizaciones puede codificarse y transmitirse en el flujo de bits. Los parámetros de la cámara asociados con las visualizaciones restantes se pueden utilizar para generar información de textura y profundidad usando la información de textura y profundidad transmitida.

[0166] De esta manera, el procedimiento de la FIG. 5 representa un ejemplo de un procedimiento que incluye recibir datos de textura para una instancia temporal de una visualización de datos de vídeo, recibir datos de profundidad correspondientes a los datos de textura para la instancia temporal de la visualización de datos de vídeo y encapsular los datos de textura y los datos de profundidad en un componente de visualización para la instancia temporal de la visualización, de modo que los datos de textura y los datos de profundidad están encapsulados dentro de un flujo de bits común.

[0167] La FIG. 6 es un diagrama de bloques que ilustra un ejemplo de un descodificador de vídeo 28, que puede descodificar una secuencia de vídeo que se codifica de manera descrita en el presente documento. El descodificador 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. 6, el descodificador de vídeo 28 corresponde al descodificador de vídeo 28 del dispositivo de destino 16. Sin embargo, en otros ejemplos, el codificador de vídeo 28 puede corresponder a un dispositivo diferente. En otros ejemplos, otras unidades (tales como, por ejemplo, otros codificadores/descodificadores (CÓDECS)) también pueden realizar técnicas similares a las del descodificador de vídeo 28.

[0168] El flujo de bits de vídeo codificado puede incluir componentes encapsulados de textura y profundidad asociados con los componentes de visualización, como se describió anteriormente. El flujo de bits puede incluir unidades NAL asociadas con cada uno de los componentes de textura y profundidad. Un indicador en la cabecera de cada unidad NAL puede indicar si la unidad NAL es una unidad NAL de componente de textura o una unidad NAL de componente de profundidad. Además, un delimitador NAL puede indicar dónde termina la unidad NAL de textura y comienza la unidad NAL de profundidad. El flujo de bits también puede incluir señales que indican dependencias predictivas de cada uno de los componentes, como se describió en detalle anteriormente. El descodificador de vídeo 28 puede determinar si una unidad NAL incluye información de textura o profundidad usando una unidad NAL de delimitador, o un indicador de profundidad en una cabecera de unidad NAL.

[0169] El descodificador de vídeo 28 incluye una unidad de descodificación por entropía 52 que descodifica por entropía los componentes del flujo de bits recibido para generar coeficientes cuantificados y los elementos sintácticos de predicción. El flujo de bits puede incluir bloques codificados de dos componentes que tienen componentes de textura y un componente de profundidad para cada ubicación de píxel para renderizar un vídeo 3D. Los elementos sintácticos de predicción pueden incluir un modo de codificación, uno o más vectores de movimiento, información que identifica una técnica de interpolación utilizada, coeficientes para uso en el filtrado de interpolación e indicación de relación predictiva entre componentes de textura y profundidad de la misma visualización y relativos a otras visualizaciones. El flujo de bits también puede estar asociado con parámetros de cámara señalados asociados con la visualización correspondiente de los componentes del flujo de bits.

[0170] Los elementos sintácticos de predicción, por ejemplo, los coeficientes, se transmiten a la unidad de predicción 55. Si se usa la predicción para codificar los coeficientes relativos a los coeficientes de un filtro fijo, o uno relativo a otro, la unidad de predicción 55 puede descodificar los elementos sintácticos para definir los coeficientes reales. Además, si la cuantificación se aplica a cualquiera de las sintaxis de predicción, la unidad de cuantificación inversa 56 también puede eliminar 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 flujo de bits codificado. Por ejemplo, el componente de profundidad puede haberse cuantificado de forma diferente que los componentes de textura. Por lo tanto, la unidad de cuantificación inversa 56 puede procesar los componentes de profundidad y textura por separado. Los coeficientes de filtro, por ejemplo, pueden codificarse mediante predicción y cuantificarse de acuerdo con esta divulgación, y en este caso, el descodificador de vídeo 28 puede utilizar la unidad de cuantificación inversa 56 para descodificar mediante predicción y descuantificar dichos coeficientes.

[0171] La unidad de predicción 55 puede generar datos de predicción basándose en los elementos sintácticos de predicción y uno o más bloques previamente descodificados que se almacenan en la memoria 62, de la misma manera que se ha descrito con detalle anteriormente con respecto a la unidad de predicción 32 del codificador de vídeo 22. En particular, la unidad de predicción 55 puede realizar una o más de las técnicas de vídeo de múltiples visualizaciones más profundidad de esta divulgación durante la compensación de movimiento y/o predicción intra para generar bloques de predicción para componentes de profundidad y componentes de textura respectivos. 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 pueden tener una precisión de un cuarto de píxel, mientras que los componentes de textura tienen una precisión de píxel entero. Así pues, el descodificador de vídeo 28 puede utilizar una o más de las técnicas de la presente divulgación en la generación de un bloque de predicción. La unidad 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. 6 para simplificar y facilitar la ilustración.

[0172] La unidad de cuantificación inversa 56 realiza la cuantificación inversa, es decir, descuantifica, los coeficientes cuantificados. El proceso de cuantificación inversa puede ser un proceso definido para la descodificación H.264 o para cualquier otra norma de descodificación. La unidad 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 predicción 55 para formar una versión reconstruida del bloque original codificado por el codificador de vídeo 22. Si se desea, también puede aplicarse un filtro de desbloqueo para filtrar los bloques descodificados con el fin de eliminar distorsiones de efecto pixelado. Los bloques de vídeo descodificados se almacenan a continuación en la memoria 62, que proporciona bloques de referencia para la compensación de movimiento posterior y también proporciona vídeo descodificado para activar el dispositivo de visualización (tal como el dispositivo 28 de la FIG. 1).

[0173] El vídeo descodificado se puede utilizar para renderizar vídeo 3D. El vídeo 3D puede comprender una visualización virtual tridimensional. La información de profundidad se puede usar para determinar un desplazamiento horizontal (disparidad horizontal) para cada píxel en el bloque. También se puede realizar la manipulación de oclusiones para generar la visualización virtual. En un ejemplo, el descodificador de vídeo 28 puede transmitir el vídeo descodificado a diferentes dispositivos de visualización, incluyendo dispositivos de visualización 2D. En este ejemplo, el descodificador de vídeo 28 puede enviar solo el componente de textura descodificado al dispositivo de visualización, pero no el componente de profundidad. En un ejemplo, el descodificador de vídeo, la pantalla o una unidad intermedia pueden generar una visualización sintetizada usando la información de textura y profundidad.

[0174] La FIG. 7 es un diagrama de flujo que ilustra un funcionamiento de ejemplo de un descodificador de vídeo coherente con esta divulgación. Por lo tanto, el proceso de la FIG. 7 puede considerarse el proceso de descodificación recíproco para el proceso de codificación de la FIG. 5. La FIG. 7 se describirá desde la perspectiva del descodificador de vídeo 28 de la FIG. 6 aunque otros dispositivos pueden realizar técnicas similares.

[0175] El descodificador de vídeo 28 puede recibir un flujo de bits que incluye componentes encapsulados de textura y profundidad de un componente de visualización (140). El descodificador de vídeo 28 puede separar el flujo de bits en los componentes de textura y profundidad, que pueden codificarse (142). El descodificador de vídeo 28 también puede obtener información señalada dentro del flujo de bits, por ejemplo, dependencias predictivas entre los datos de textura y profundidad de la misma visualización y relativa a los datos de textura y profundidad de otras visualizaciones y parámetros de la cámara asociados con el componente de visualización.

[0176] El descodificador de vídeo 28 determina un modo de predicción para determinar a partir de qué bloque de predicción se determinaron los coeficientes residuales en el codificador (tal como el codificador de vídeo 22) (144). Basándose en una determinación del modo de predicción, el descodificador de vídeo 28 puede calcular datos de predicción para los bloques codificados (146). Usando el modo de predicción determinado y los datos de predicción, el descodificador de vídeo 28 puede descodificar cada uno de los componentes de textura y profundidad (148). Para descodificar los componentes, el descodificador 28 de vídeo puede usar las referencias y relaciones de predicción señaladas apropiadas, y usar funciones de descodificación de vídeo. Por ejemplo, el descodificador de vídeo 28 puede descodificar los coeficientes residuales, que pueden cuantificarse inversamente (por ejemplo, mediante la unidad de cuantificación inversa 56) y transformarse inversamente (por ejemplo, mediante la unidad de transformada inversa 58).

[0177] Usando los coeficientes residuales descodificados, el descodificador de vídeo 28 puede combinar los coeficientes residuales con los datos de predicción para formar un bloque reconstruido (150). Desde el bloque

reconstruido, el descodificador 28 de vídeo puede generar una trama de la imagen que tiene componentes de textura y profundidad.

5 **[0178]** Usando la trama que tiene componentes de textura y el mapa de profundidad, el descodificador de vídeo 28 puede generar una visualización virtual tridimensional de la trama (152). La trama puede combinarse con otras tramas descodificadas para generar una visualización virtual tridimensional de una imagen. En algunos ejemplos, el descodificador de vídeo 28 no genera la visualización virtual tridimensional, sino que reenvía la trama que tiene los componentes de textura y un mapa de profundidad a un dispositivo de renderización externo.

10 **[0179]** De esta manera, el procedimiento de la FIG. 7 representa un ejemplo de un procedimiento que incluye recibir datos de profundidad y datos de textura encapsulantes de flujo de bits comunes en un componente de visualización para una instancia temporal de una visualización de datos de vídeo, en el que los datos de profundidad corresponden a los datos de textura para la instancia temporal de la visualización de datos de vídeo, y separar los datos de textura y los datos de profundidad para la instancia temporal de la visualización de datos de vídeo.

15 **[0180]** 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, como una o más instrucciones o código, pueden almacenarse en, y transmitirse por, un medio legible por ordenador, 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 unos medios de comunicación que incluyen cualquier medio que facilite la transferencia de un programa informático desde un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador pueden corresponder en general a (1) unos medios de almacenamiento tangibles legibles por ordenador que son no transitorios, o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser medios disponibles cualesquiera a los que se puede acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en esta divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

20 **[0181]** A modo de ejemplo, y no de manera limitativa, tales medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento de disco óptico, almacenamiento de disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que pueda usarse para almacenar código de programa deseado en forma de instrucciones o estructuras de datos y al que pueda accederse mediante un ordenador. Además, cualquier conexión recibe debidamente la denominación de medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, un servidor u otro origen remoto usando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas se incluyen en la definición de medio. Sin embargo, debería entenderse que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que, en cambio, se orientan a medios de almacenamiento tangibles no transitorios. Los discos, como se usan en el presente documento, incluyen un disco compacto (CD), un disco láser, un disco óptico, un disco versátil digital (DVD), un disco flexible y un disco Blu-ray donde algunos discos reproducen usualmente los datos magnéticamente, mientras que otros discos reproducen los datos ópticamente con láseres. Las combinaciones de los anteriores también deben incluirse dentro del alcance de los medios legibles por ordenador.

25 **[0182]** 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 por campo (FPGA) u otra circuitería lógica integrada o discreta equivalente. Por consiguiente, el término «procesador», como se utiliza en el presente documento, puede referirse a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento puede proporcionarse dentro de módulos de hardware y/o software dedicados configurados para la codificación y la descodificación, o incorporarse en un códec combinado. Asimismo, las técnicas podrían implementarse por completo en uno o más circuitos o elementos lógicos.

30 **[0183]** Las técnicas de la presente divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluidos un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). Diversos componentes, módulos o unidades se describen en esta divulgación para enfatizar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no requieren necesariamente su realización mediante diferentes unidades de hardware. En cambio, como se ha descrito anteriormente, diversas unidades 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 adecuados.

[0184] Se han descrito diversos ejemplos de la divulgación. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.

REIVINDICACIONES

1. Un procedimiento para encapsular datos de vídeo; el procedimiento que comprende:

5 recibir (102) un componente de visualización de textura que comprende datos de textura para una instancia temporal de una visualización de datos de vídeo;

recibir (106) un componente de visualización de profundidad que comprende datos de profundidad correspondientes a los datos de textura para la instancia temporal de la visualización de datos de vídeo; y

10 encapsular, en una primera unidad de capa de abstracción de red, NAL, de capa de códec de vídeo, VCL, el componente de visualización de textura y encapsular, en una segunda unidad VCL NAL después de la primera unidad VCL NAL, el componente de visualización de profundidad, las primera y la segunda unidades VCL NAL que forman un componente de visualización para la instancia temporal de la visualización (74-1; 74-2;... 74-M), de modo que el componente de visualización de textura y el componente de visualización de profundidad se encapsulan dentro de unidades VLC NAL separadas dentro de un flujo de bits común;

señalar el componente de visualización;

20 señalar, en una cabecera de la primera unidad VCL NAL, una primera indicación de que el primer tipo de unidad VCL NAL incluye los datos de textura, estando el campo asociado con un tipo de unidad NAL de la primera unidad NAL; y

25 señalar, en una cabecera de la segunda unidad VCL NAL, una segunda indicación de que la segunda unidad VCL NAL incluye los datos de profundidad, estando el campo asociado con un tipo de unidad NAL de la segunda unidad NAL.

2. El procedimiento según la reivindicación 1, que comprende además:

30 formar el flujo de bits común para incluir el componente de visualización de textura del componente de visualización seguido consecutivamente por el componente de visualización de profundidad del componente de visualización; y

35 formar una unidad NAL delimitadora,

en el que el encapsulado comprende proporcionar la unidad NAL delimitadora en una unidad de acceso del flujo de bits común entre la primera unidad VCL NAL y la segunda unidad VCL NAL.

3. El procedimiento según la reivindicación 1, que comprende además señalar los parámetros de la cámara para una o más cámaras relacionadas con visualizaciones de datos de vídeo en el flujo de bits, en el que la señalización de los parámetros de la cámara comprende:

45 determinar los parámetros de cámara intrínsecos compartidos por las visualizaciones correspondientes a la una o más cámaras, comprendiendo los parámetros de cámara intrínsecos uno o más de la distancia focal y el desplazamiento del punto principal;

determinar los parámetros de cámara extrínsecos compartidos por las visualizaciones correspondientes a la una o más cámaras;

50 determinar los parámetros de cámara específicos de visualización que incluyen las ubicaciones horizontales del mundo real de la una o más cámaras; y

55 formar una estructura de datos a nivel de secuencia que incluye valores indicativos de los parámetros de cámara intrínsecos, los parámetros de cámara extrínsecos y los parámetros de cámara específicos de visualización.

4. El procedimiento según la reivindicación 1, que comprende además datos de señalización que representan la conversión de valores de profundidad (z) del mundo real a valores de profundidad representados en los datos de profundidad como tramas codificadas, en el que los datos que representan la conversión comprenden datos que representan al menos uno de una indicación de una función lineal para convertir los valores de profundidad (z) del mundo real en valores de profundidad del componente de visualización de profundidad, una indicación de una función lineal inversa para convertir los valores de profundidad (z) del mundo real en valores de profundidad del componente de visualización de profundidad, datos que representan un rango de los valores de profundidad (z) del mundo real, y una indicación de una tabla de búsqueda utilizada para señalar los valores de profundidad (z) del mundo real, en el que los datos que representan la conversión se forman en una unidad NAL que comprende un valor de tipo de unidad NAL único y los datos que representan la conversión,

comprendiendo además el procedimiento la señalización de un conjunto de parámetros de visualización en el flujo común en una o más unidades de acceso, en el que el conjunto de parámetros de visualización incluye información indicativa del rango de los valores de la profundidad (z) del mundo real.

5 **5.** Un dispositivo para procesar datos de vídeo que comprende:

medios (22) para recibir un componente de visualización de textura (4) que comprende datos de textura para una instancia temporal de una visualización de datos de vídeo;

10 medios (22) para recibir un componente de visualización de profundidad (6) que comprende datos de profundidad correspondientes a los datos de textura para la instancia temporal de la visualización de datos de vídeo; y

15 medios (22) para encapsular, en una primera unidad de capa de abstracción de red, NAL, de capa de códec de vídeo, VCL, el componente de visualización de textura y encapsular, en una segunda unidad VCL NAL después de la primera unidad VCL NAL, el componente de visualización de profundidad, con la primera y segunda unidades VCL NAL que forman un componente de visualización (8) para la instancia temporal de la visualización, de modo que el componente de visualización de textura (4) y el componente de visualización de profundidad (6) están encapsulados dentro de unidades VCL NAL separadas dentro de un flujo de bits común;

medios para señalar el componente de visualización; y

25 medios (22) para señalar, en una cabecera de la primera unidad VCL NAL, una primera indicación de que el primer tipo de unidad VCL NAL incluye datos de textura, estando el campo asociado con un tipo de unidad NAL de la primera unidad NAL y para señalar, en una cabecera de la segunda unidad VCL NAL, una segunda indicación de que la segunda unidad VCL NAL incluye los datos de profundidad, estando el campo asociado con un tipo de unidad NAL de la segunda unidad NAL.

30 **6.** El dispositivo según la reivindicación 5, que comprende además:

medios para formar una unidad NAL delimitadora, y

35 en el que el encapsulado comprende proporcionar la unidad NAL delimitadora en una unidad de acceso del flujo de bits común entre la primera unidad VCL NAL y la segunda unidad VCL NAL.

40 **7.** El dispositivo según la reivindicación 5, que comprende además medios para señalar parámetros de cámara para una o más cámaras relacionadas con visualizaciones de datos de vídeo en el flujo de bits, en el que los medios para señalar los parámetros de cámara comprenden:

medios para determinar los parámetros de cámara intrínsecos compartidos por las visualizaciones correspondientes a la una o más cámaras, comprendiendo los parámetros de cámara intrínsecos uno o más de distancia focal y el desplazamiento del punto principal;

45 medios para determinar parámetros de cámara extrínsecos compartidos por las visualizaciones correspondientes a la una o más cámaras;

medios para determinar parámetros de cámara específicos de visualización que incluyen ubicaciones horizontales del mundo real de la una o más cámaras; y

50 medios para formar una estructura de datos a nivel de secuencia que incluye valores indicativos de los parámetros de cámara intrínsecos, los parámetros de cámara extrínsecos y los parámetros de cámara específicos de visualización.

55 **8.** El dispositivo según la reivindicación 5, que comprende además medios para señalar datos que representan conversión de valores de profundidad (z) del mundo real a valores de profundidad representados en los datos de profundidad como tramas codificadas, en el que los datos que representan la conversión comprenden datos que representan al menos uno de una indicación de una función lineal para convertir los valores de profundidad (z) del mundo real en valores de profundidad del componente de visualización de profundidad, una indicación de una función lineal inversa para convertir los valores de profundidad (z) del mundo real en valores de profundidad del componente de visualización de profundidad, datos que representan un rango de los valores de profundidad (z) del mundo real, y una indicación de una tabla de búsqueda utilizada para señalar los valores de profundidad (z) del mundo real, en el que los medios para señalar los datos que representan la conversión comprenden medios para formar una unidad NAL que comprende un valor de tipo de unidad NAL único y los datos que representan la conversión, y medios para señalar un conjunto de parámetros de visualización en el

flujo común en una o más unidades de acceso, en el que el conjunto de parámetros de visualización incluye información indicativa del rango de los valores de profundidad (z) del mundo real.

- 5 **9.** Un procedimiento para desencapsular datos de vídeo, comprendiendo el procedimiento:
- recibir (140) un encapsulado de flujo de bits común, en una primera unidad de capa de abstracción de red, NAL, de capa de códec de vídeo, VCL, un componente de visualización de textura y en una segunda unidad VCL NAL después de la primera unidad VCL NAL un componente de visualización de profundidad, con la primera y la segunda unidades VCL NAL formando un componente de visualización para una instancia temporal de la visualización de datos de vídeo, en el que el componente de visualización de textura comprende datos de textura para la instancia temporal de una visualización y el componente de visualización de profundidad comprende datos de profundidad correspondientes a los datos de textura para la instancia temporal de visualización de datos de vídeo; y
- 10 separar (142) el componente de visualización de textura y el componente de visualización de profundidad para la instancia temporal de la visualización de datos de vídeo utilizando una indicación de señal recibida en las cabeceras de unidad NAL de las unidades NAL de si una unidad NAL incluye información de textura o profundidad.
- 15 **10.** El procedimiento según la reivindicación 9, que comprende además desencapsular una unidad NAL delimitadora incluida en el flujo de bits común entre la primera unidad VCL NAL y la segunda unidad VCL NAL.
- 20 **11.** Un dispositivo para procesar datos de vídeo que comprende:
- 25 medios (26; 27) para recibir un de flujo de bits común que encapsula, en una primera unidad VCL NAL, un componente visualización de textura y, en una segunda unidad VCL NAL después de la primera unidad VCL NAL, un componente visualización de profundidad, con la primera y segunda unidades VCL NAL que forman un componente de visualización (8) para una instancia temporal de la visualización de datos de vídeo, en el que el componente de visualización de textura comprende datos de textura para la instancia temporal de una visualización y el componente de visualización de profundidad comprende datos de profundidad correspondientes a los datos de textura para la instancia temporal de la visualización de datos de vídeo; y
- 30 medios (28) para separar el componente de visualización de textura (4) y el componente de visualización de profundidad (6) para la instancia temporal de la visualización de datos de vídeo usando una indicación de señal recibida en las cabeceras de unidad NAL de las unidades NAL de si una unidad NAL incluye información de textura o profundidad.
- 35 **12.** El dispositivo de la reivindicación 11, que comprende además medios para desencapsular una unidad NAL delimitadora incluida en el flujo de bits común entre la primera unidad VCL NAL y la segunda unidad VCL NAL.
- 40 **13.** Un producto de programa informático que comprende un medio de almacenamiento legible por ordenador que tiene almacenadas en el mismo instrucciones que, cuando se ejecutan, hacen que un procesador de un dispositivo de codificación de vídeo lleve a cabo el procedimiento de cualquiera de las reivindicaciones 1 a 4 o 9 a 10.
- 45

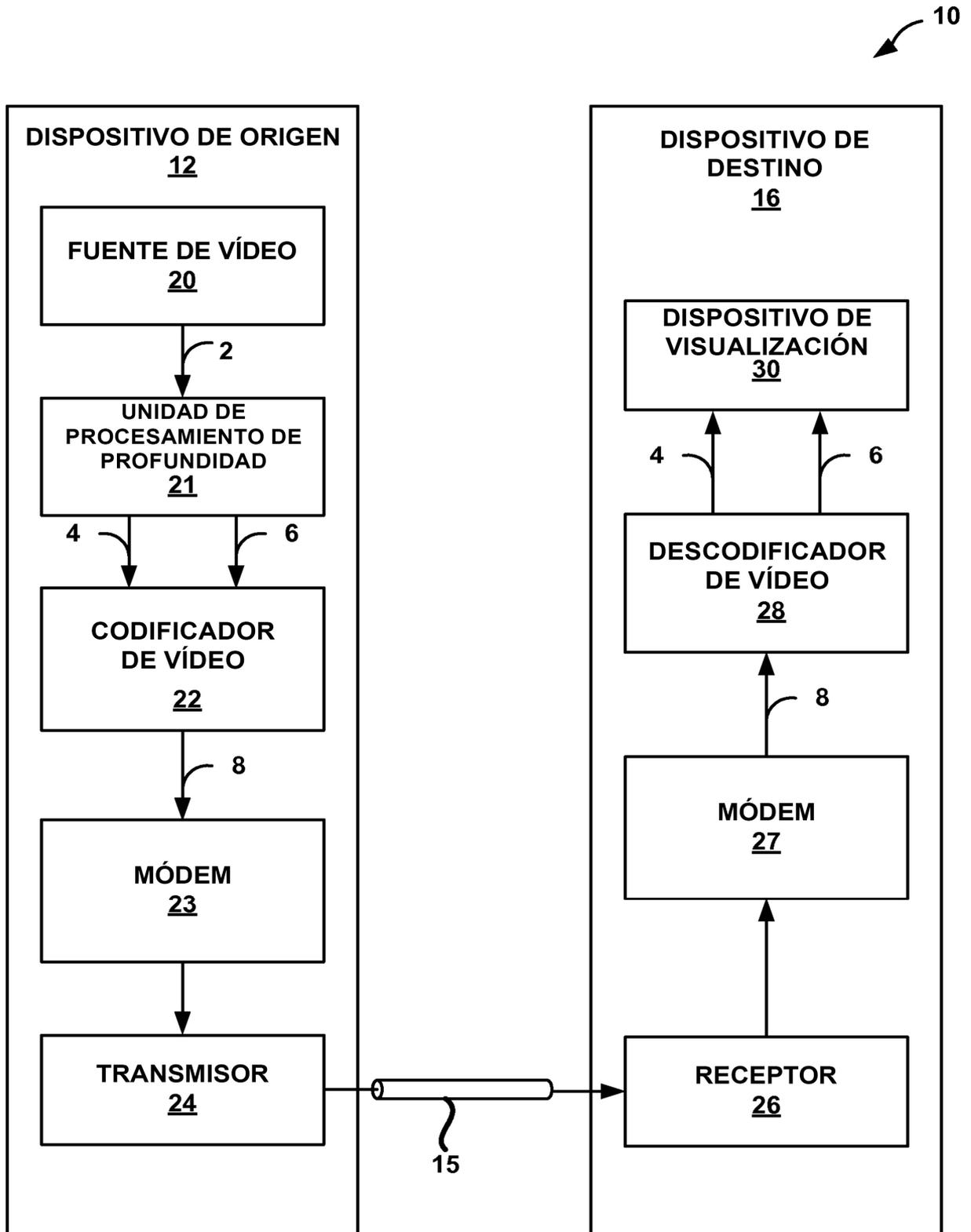


FIG. 1

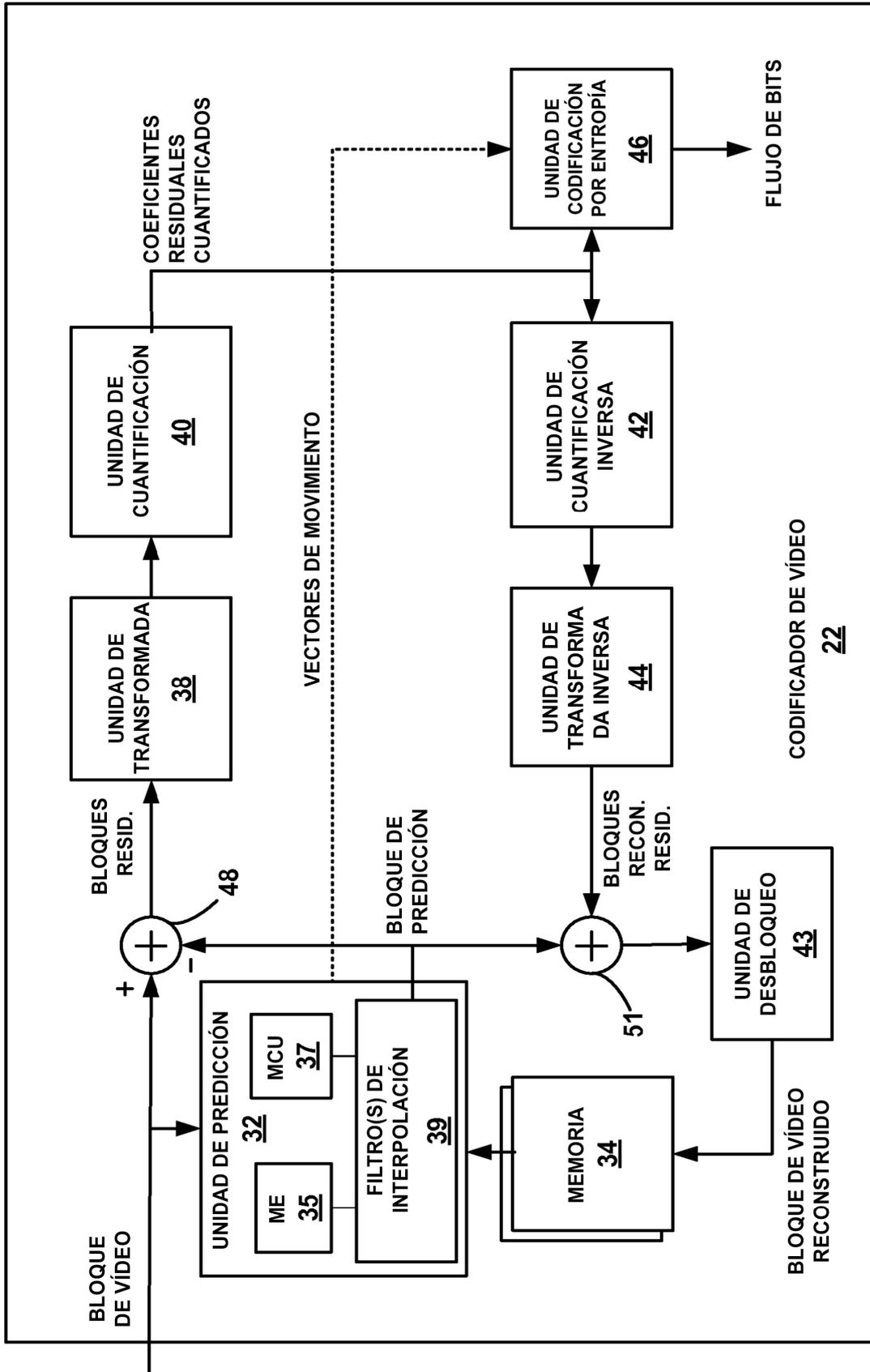


FIG. 2

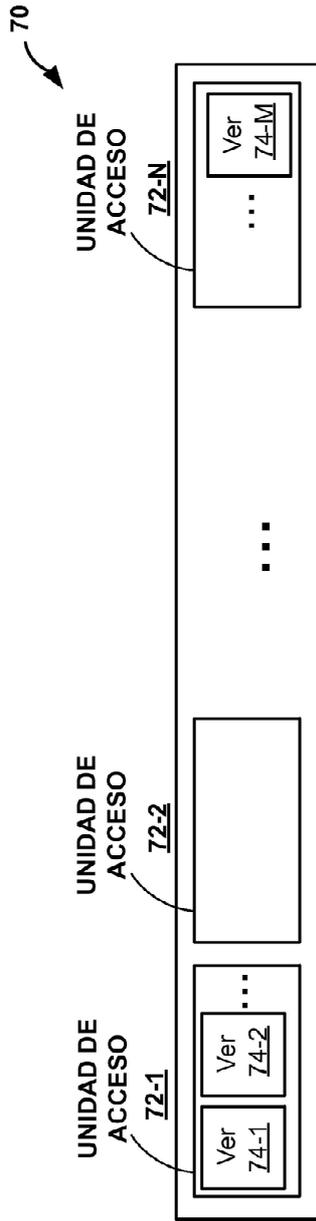


FIG. 3A

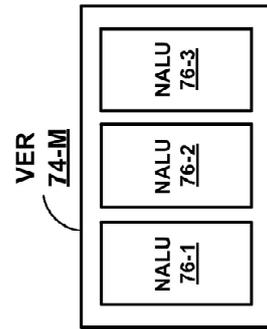


FIG. 3B

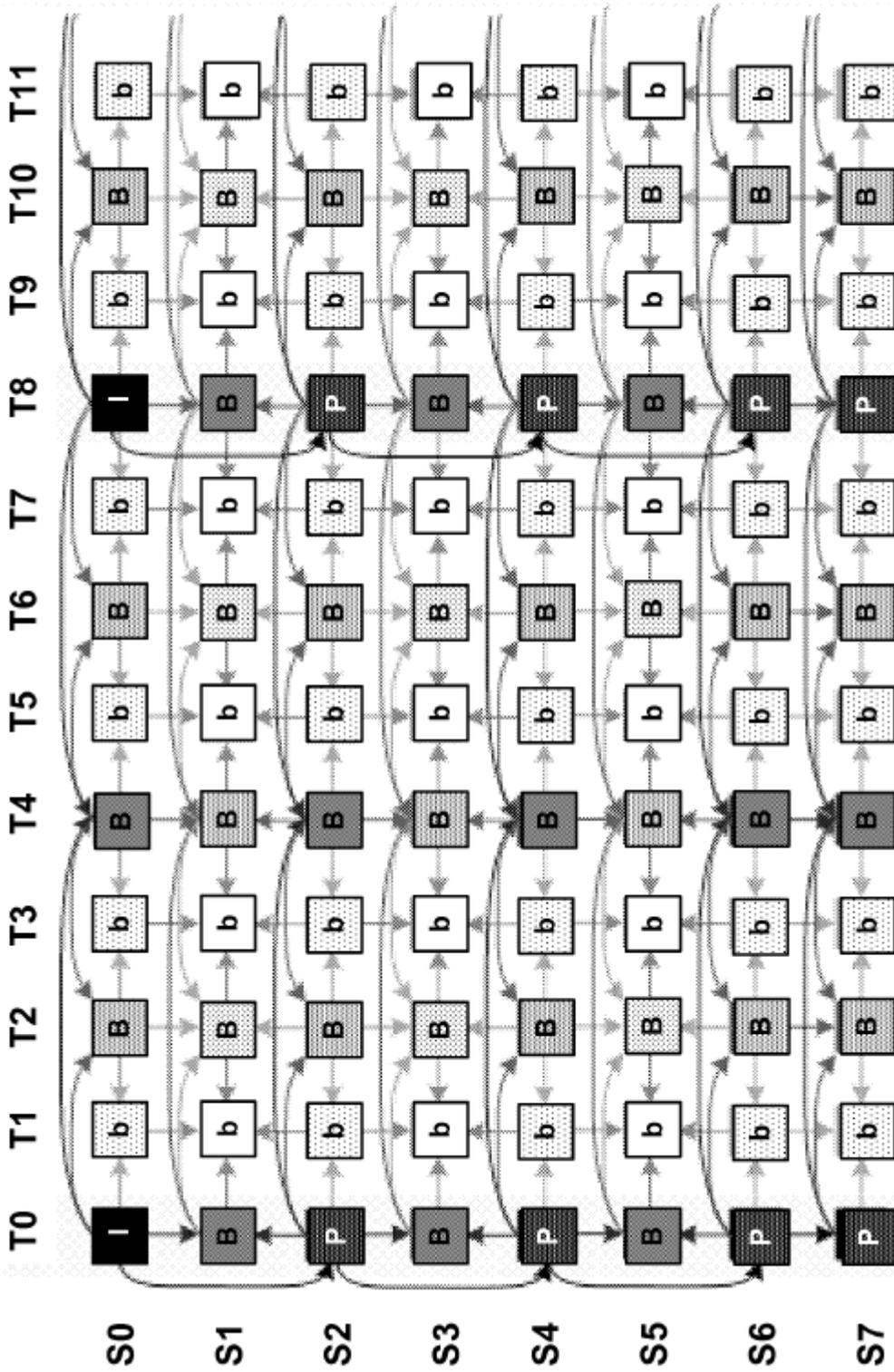


FIG. 4A

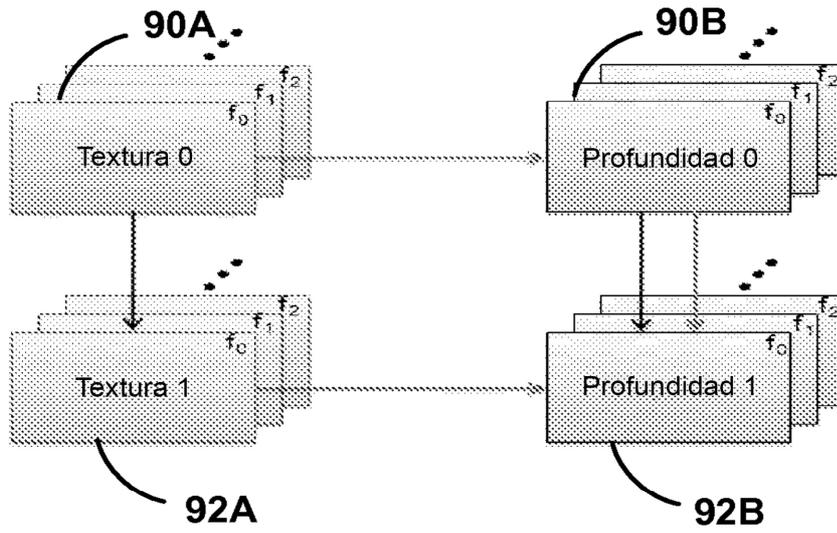


FIG. 4B

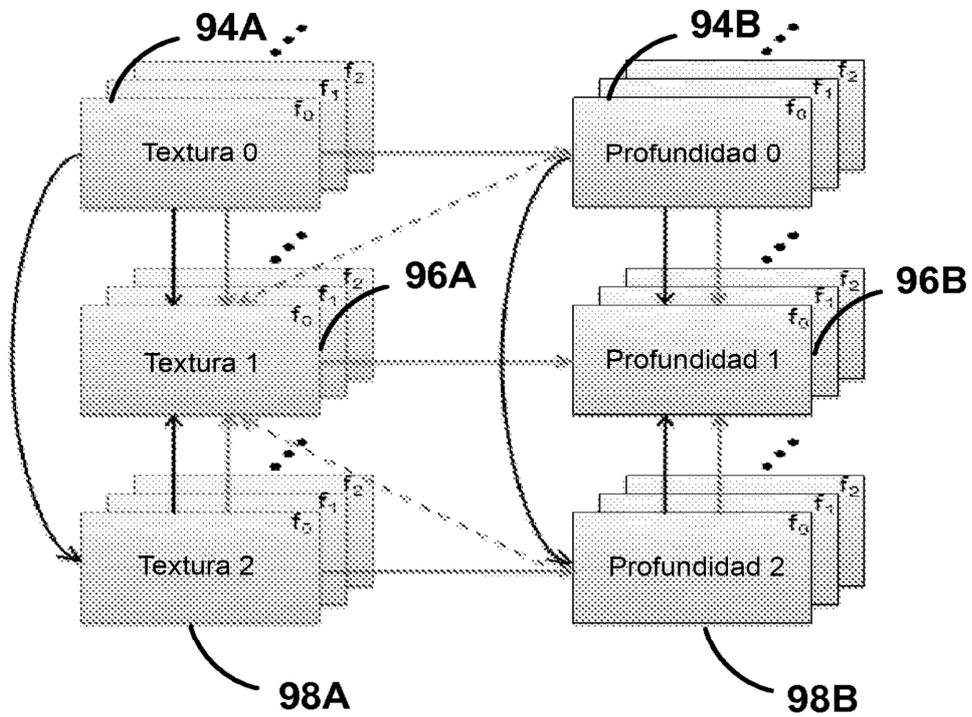


FIG. 4C

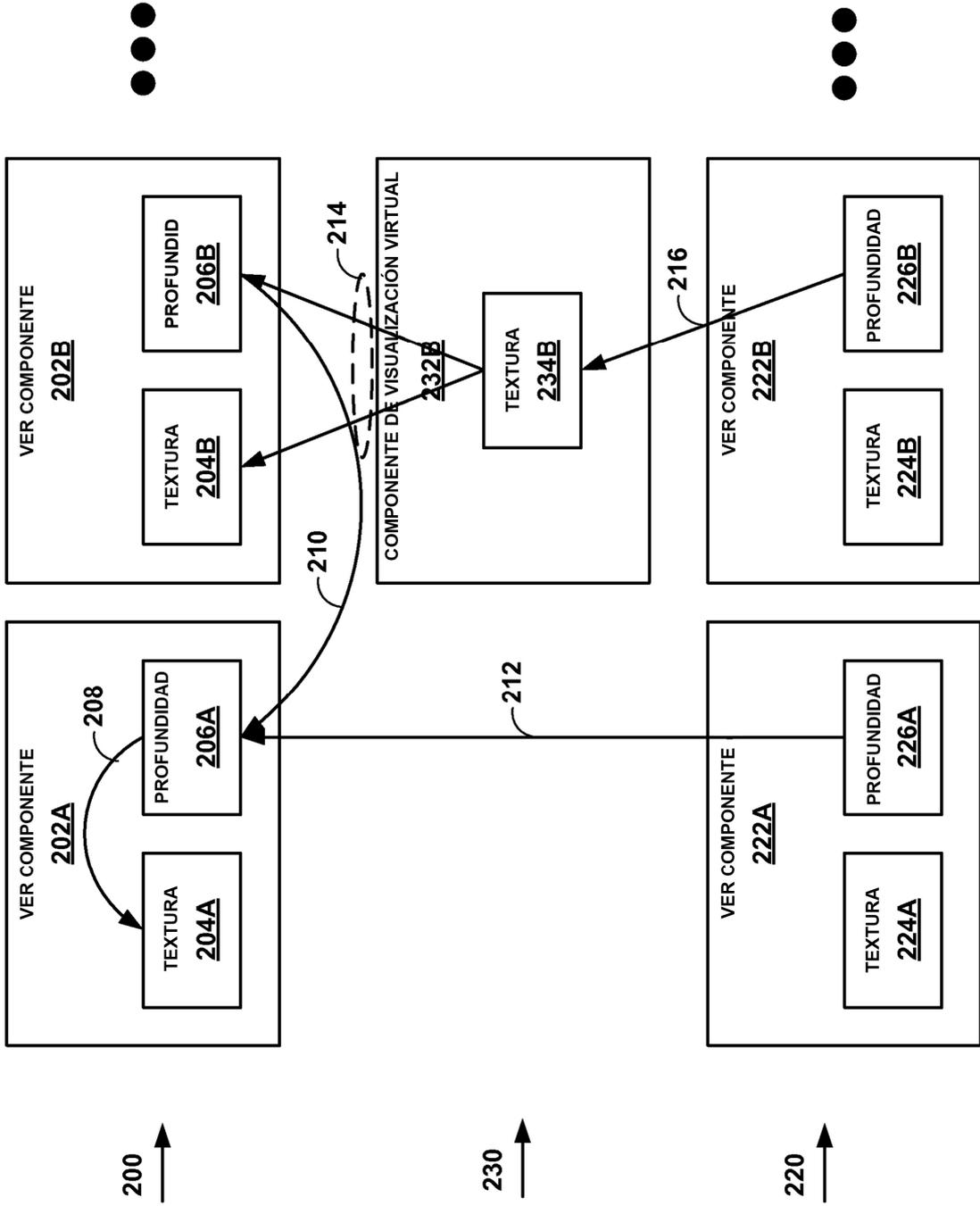


FIG. 4D

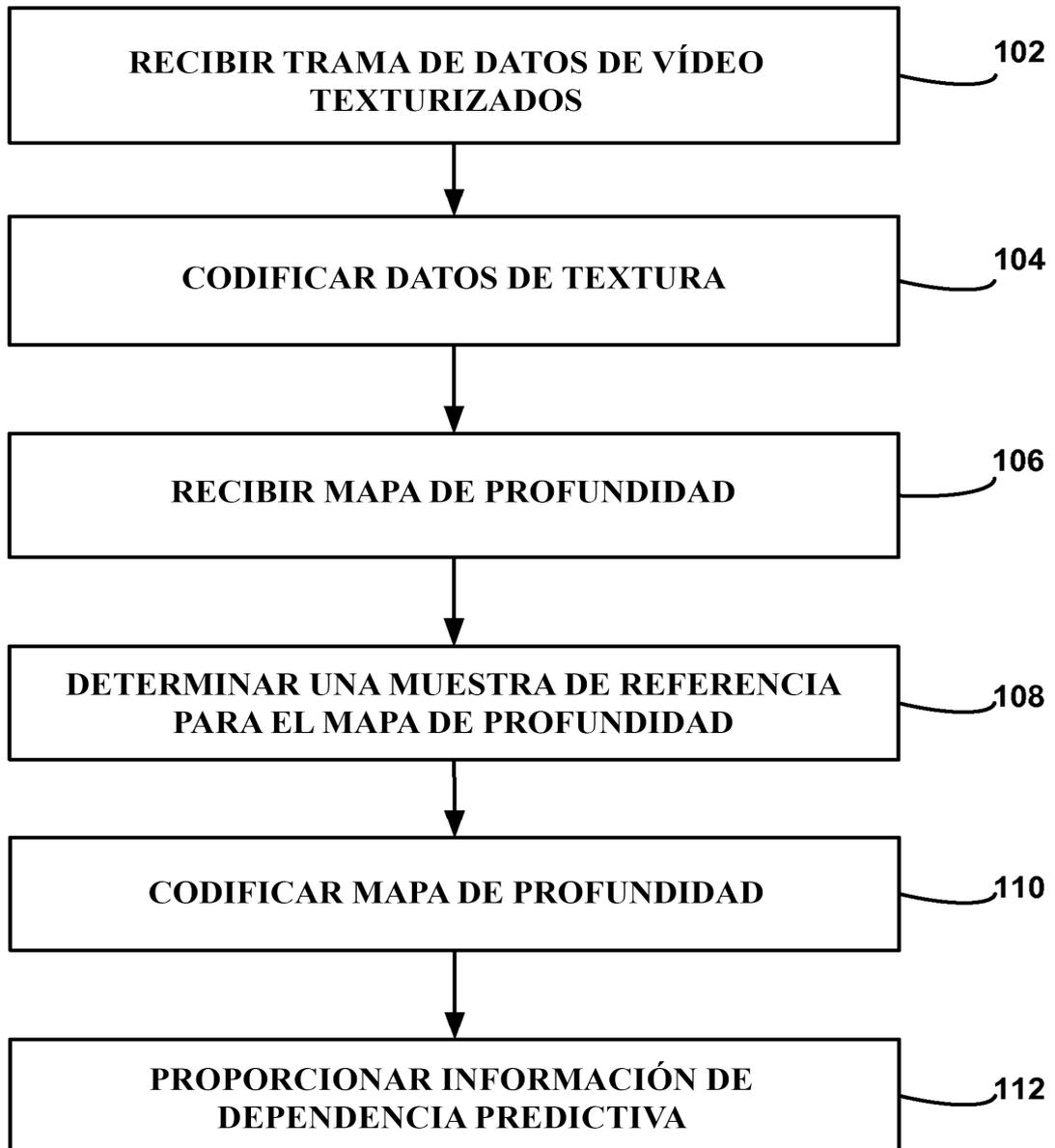


FIG. 5

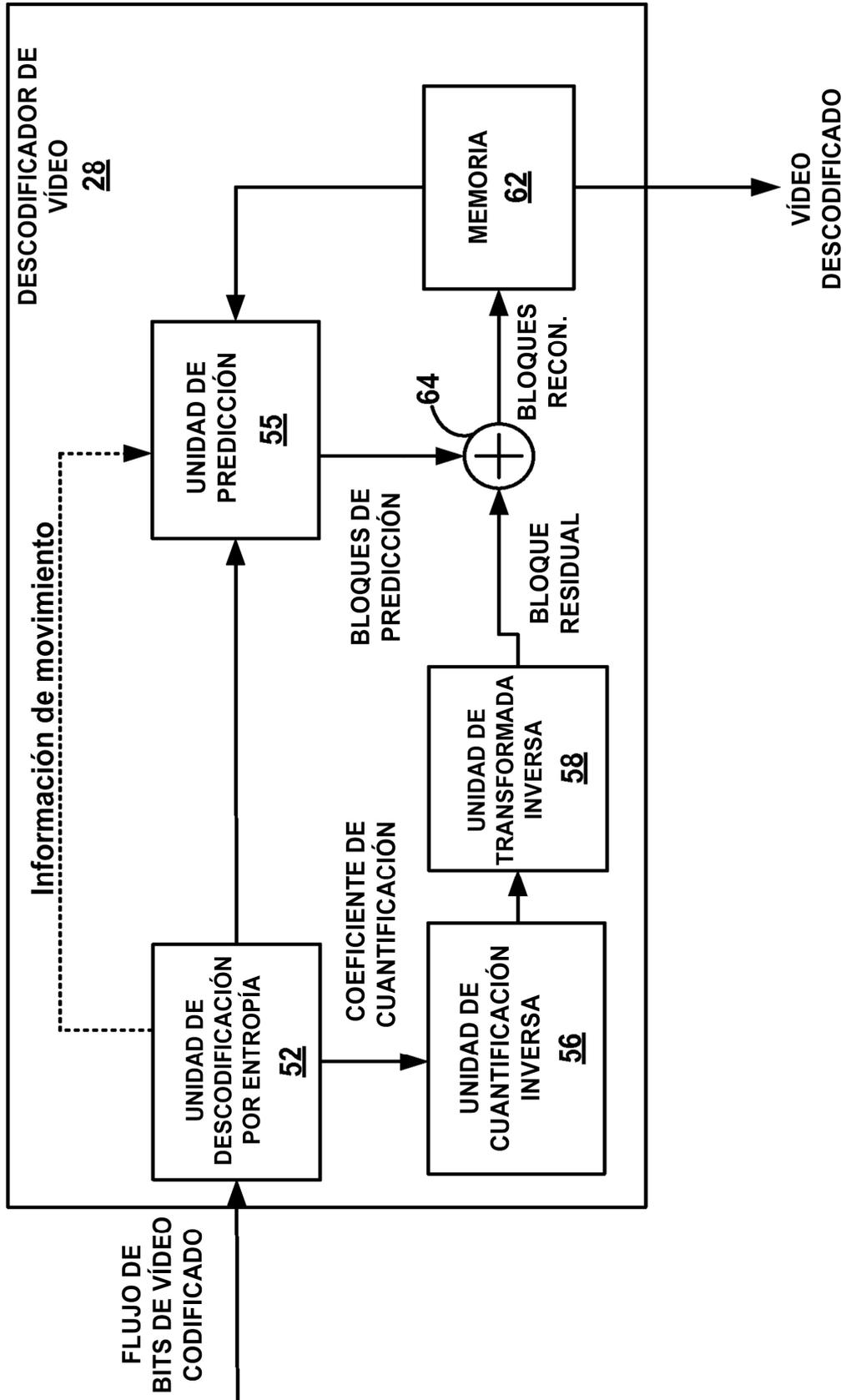


FIG. 6

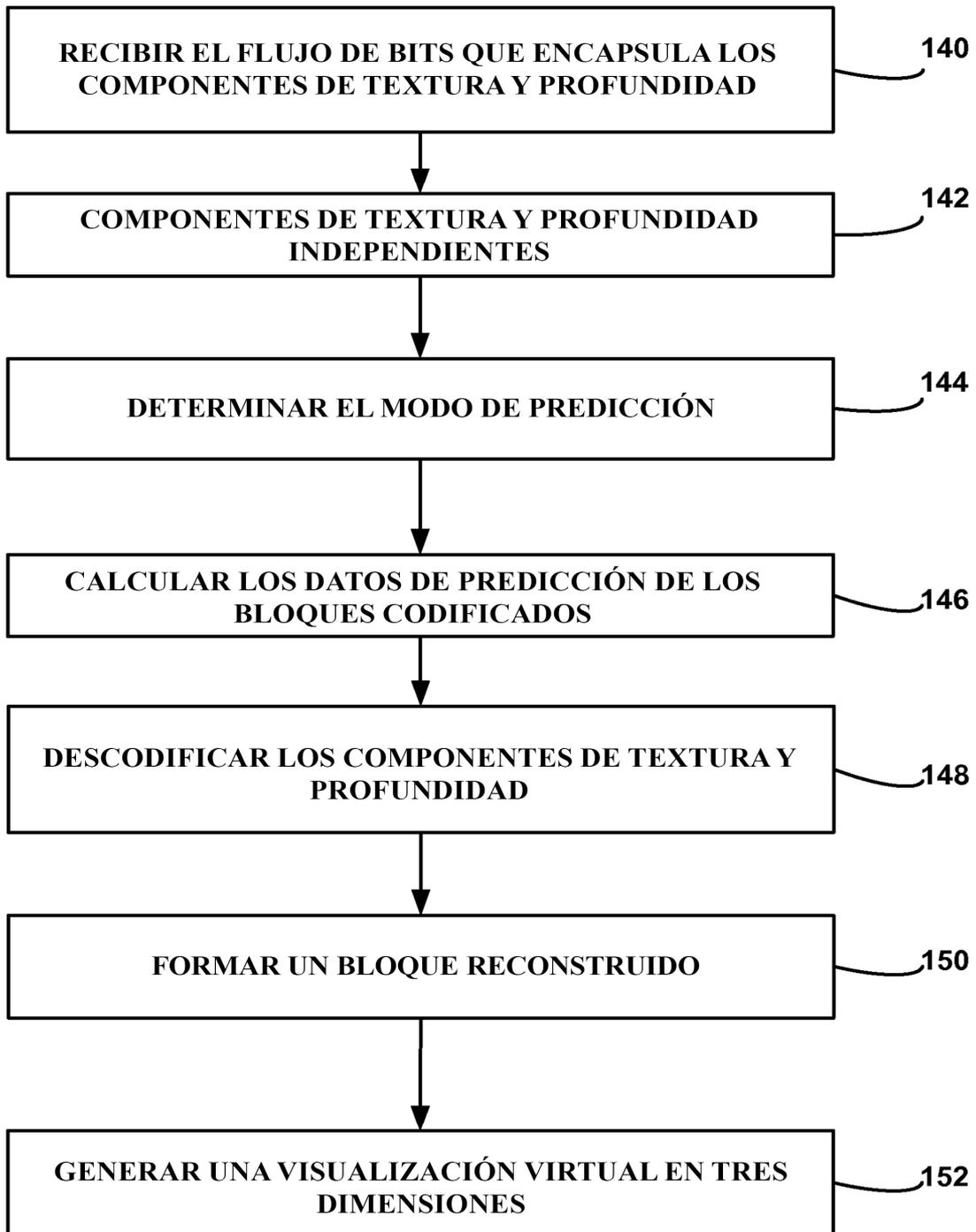


FIG. 7