

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 745 282**

51 Int. Cl.:

H04N 19/597 (2014.01)

H04N 19/11 (2014.01)

H04N 19/17 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **15.10.2014 PCT/US2014/060732**

87 Fecha y número de publicación internacional: **23.04.2015 WO15057861**

96 Fecha de presentación y número de la solicitud europea: **15.10.2014 E 14795713 (8)**

97 Fecha y número de publicación de la concesión europea: **12.06.2019 EP 3058738**

54 Título: **Extensión del patrón de minicuñas para la intracodificación de profundidad**

30 Prioridad:

15.10.2013 US 201361891203 P

14.10.2014 US 201414513839

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

28.02.2020

73 Titular/es:

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

72 Inventor/es:

**ZHAO, XIN;
ZHANG, LI;
CHEN, YING y
KARCZEWICZ, MARTA**

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 745 282 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

DESCRIPCIÓN

Extensión del patrón de minicuñas para la intracodificación de profundidad

- 5 **[0001]** Esta solicitud reivindica el beneficio de la solicitud provisional estadounidense número 61/891.203 presentada el 15 de octubre de 2013.

CAMPO TÉCNICO

- 10 **[0002]** Esta divulgación se refiere a la codificación de vídeo.

ANTECEDENTES

- 15 **[0003]** Las capacidades del vídeo digital pueden incorporarse en una amplia gama de dispositivos, que incluye televisores digitales, sistemas de difusión digital directa, sistemas de difusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, ordenadores de tableta, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos celulares o de radio por satélite, los denominados "teléfonos inteligentes", dispositivos de videoconferencia, dispositivos de transmisión de vídeo en tiempo real y similares. Los dispositivos de vídeo digital implementan técnicas de compresión de vídeo, tales como las descritas en las normas definidas por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, parte 10, codificación de vídeo avanzada (AVC), la norma de codificación de vídeo de alta eficacia (HEVC) actualmente en desarrollo y las ampliaciones de dichas normas. Los dispositivos de vídeo pueden transmitir, recibir, codificar, decodificar y/o almacenar información de vídeo digital con más eficacia implementando dichas técnicas de compresión de vídeo.

- 25 **[0004]** Las técnicas de compresión de vídeo realizan predicción espacial (intraimagen) y/o predicción temporal (interimagen) para reducir o eliminar la redundancia intrínseca a las secuencias de vídeo. Para la codificación de vídeo basada en bloques, un fragmento de vídeo (es decir, una imagen o una parte de una imagen) se puede dividir en bloques de vídeo, que también se pueden denominar bloques arbolados, unidades de codificación (CU) y/o nodos de codificación. Los bloques de vídeo en un fragmento intracodificado (I) de una imagen se codifican usando predicción espacial con respecto a muestras de referencia en bloques próximos en la misma imagen. Los bloques de vídeo en un fragmento intercodificado (P o B) de una imagen pueden usar la predicción espacial con respecto a muestras de referencia en bloques próximos en la misma imagen o la predicción temporal con respecto a muestras de referencia en otras imágenes de referencia.

- 35 **[0005]** La predicción espacial o temporal da como resultado un bloque predictivo para un bloque a codificar. Los datos residuales representan diferencias de píxeles entre el bloque original a codificar y el bloque predictivo. Un bloque intercodificado se codifica de acuerdo a un vector de movimiento que apunta a un bloque de muestras de referencia que forman el bloque predictivo, y los datos residuales que indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intracodificado se codifica de acuerdo a una modalidad de intracodificación y a los datos residuales. Para una mayor compresión, los datos residuales pueden transformarse desde el dominio espacial a un dominio de transformación, dando como resultado unos coeficientes de transformación residuales, que posteriormente se pueden cuantizar. Los coeficientes de transformación cuantizados, dispuestos inicialmente en una formación bidimensional, pueden explorarse con el fin de generar un vector unidimensional de coeficientes de transformación, y puede aplicarse codificación por entropía para lograr aún más compresión.

- 40 **[0006]** El documento WO 2013/068566 A1 describe un procedimiento de codificación de partición adaptativa. El artículo de Kawamura et al, en "3D-CE6.h related: Calculation Sample Reduction for Depth Modelling Mode 1 and 3 [Referido a 3D-CE6.h: Reducción de muestra de cálculo para la modalidad de modelado en profundidad 1 y 3]", 103. CONFERENCIA DEL MPEG; 21-1-2013 a 25-1-2013; GINEBRA; (GRUPO DE EXPERTOS EN IMÁGENES EN MOVIMIENTO O ISO/IEC JTC1/SC29/WG11),, (20130117), n° m28288, propone simplificar la Modalidad de modelado en profundidad (DMM) 1 y 3, omitiendo el cálculo de la región sin borde, basándose en el patrón de minicuñas.

SUMARIO

- 55 **[0007]** Las técnicas de esta divulgación generalmente se refieren a la obtención de un patrón de partición para un bloque intrapredicho de datos de vídeo basándose en un patrón de partición asociado a bloques de menor tamaño. En algunos ejemplos, el bloque intrapredicho puede ser un bloque de un mapa de profundidad, y el patrón de partición puede ser un patrón de partición basado en líneas que divide el subbloque en dos particiones divididas por una línea lineal que biseca el subbloque.

- 60 **[0008]** En un ejemplo, la divulgación describe un procedimiento para decodificar datos de vídeo, comprendiendo el procedimiento construir una lista de patrones de partición que incluye uno o más patrones de partición asociados a un bloque de un primer tamaño, recibir un índice de la lista de patrones de partición, determinar un patrón de particiones asociado al bloque del primer tamaño, basándose en el índice recibido de la lista de patrones de partición, determinar un patrón de partición para un bloque de profundidad de un segundo tamaño basándose en el patrón de partición

determinado asociado al bloque del primer tamaño, en donde el segundo tamaño es mayor que el primer tamaño, en donde el bloque del primer tamaño está ubicado dentro del bloque de profundidad del segundo tamaño y en donde el bloque de profundidad es un bloque de un mapa de profundidad en un entorno de codificación de vídeo tridimensional (3D) y decodificación por intrapredicción del bloque de profundidad, basándose en el patrón de partición determinado para el bloque de profundidad sin la decodificación por intrapredicción del bloque del primer tamaño usando el patrón de partición determinado, asociado al bloque del primer tamaño, en donde el bloque del primer tamaño se decodifica por intrapredicción como parte de la intrapredicción del bloque de profundidad.

[0009] En un ejemplo, la divulgación describe un procedimiento para codificar datos de vídeo, comprendiendo el procedimiento construir una lista de patrones de partición que incluye uno o más patrones de partición asociados a un bloque de un primer tamaño, determinar un patrón de partición asociado al bloque del primer tamaño basándose en la lista de patrones de partición, determinar un patrón de partición para un bloque de profundidad de un segundo tamaño basándose en el patrón de partición determinado asociado al bloque del primer tamaño, en donde el segundo tamaño es mayor que el primer tamaño, en donde el bloque del primer tamaño se ubica dentro del bloque de profundidad del segundo tamaño y en donde el bloque de profundidad es un bloque de un mapa de profundidad en un entorno de codificación de vídeo tridimensional (3D), codificar por intrapredicción el bloque de profundidad basándose en el patrón de partición determinado para el bloque de profundidad sin la codificación por intrapredicción del bloque del primer tamaño, utilizar el patrón de partición determinado asociado al bloque del primer tamaño, en donde el bloque del primer tamaño se codifica por intrapredicción como parte de la intrapredicción del bloque de profundidad, y señalar un índice de la lista de patrones de partición construida, que identifica el patrón de partición determinado asociado al bloque del primer tamaño, en donde un decodificador de vídeo usa el índice para decodificar por intrapredicción el bloque de profundidad.

[0010] En un ejemplo, la divulgación describe un dispositivo para la decodificación de vídeo, comprendiendo el dispositivo una memoria de datos de vídeo que almacena uno o más patrones de partición asociados a un bloque de un primer tamaño, y un codificador de vídeo que comprende uno o más procesadores. El codificador de vídeo está configurado para construir una lista de patrones de partición que incluye los uno o más patrones de partición asociados al bloque del primer tamaño, determinar un patrón de partición asociado al bloque del primer tamaño a partir de los uno o más patrones de partición almacenados, basándose en la lista de patrones de partición, en donde el patrón de partición en la lista de patrones de partición se identifica mediante un índice de la lista de patrones de partición; determinar un patrón de partición para un bloque de profundidad de un segundo tamaño, basándose en el patrón de partición determinado asociado al bloque del primer tamaño, en donde el segundo tamaño es mayor que el primer tamaño, en donde el bloque del primer tamaño está ubicado dentro del bloque de profundidad del segundo tamaño y en donde el bloque de profundidad es un bloque de un mapa de profundidad en un entorno de codificación de vídeo tridimensional (3D); decodificar por intrapredicción el bloque de profundidad basándose en el patrón de partición determinado para el bloque de profundidad sin decodificar por intrapredicción el bloque del primer tamaño utilizando el patrón de partición determinado asociado al bloque del primer tamaño, en donde el bloque del primer tamaño se decodifica por intrapredicción como parte de la intrapredicción del bloque de profundidad.

[0011] En un ejemplo, la divulgación describe un dispositivo para codificación de vídeo, comprendiendo el dispositivo una memoria de datos de vídeo que almacena uno o más patrones de partición asociados a un bloque de un primer tamaño, y un codificador de vídeo que comprende uno o más procesadores. El codificador de vídeo está configurado para construir una lista de patrones de partición que incluye los uno o más patrones de partición asociados al bloque del primer tamaño, determinar un patrón de partición asociado al bloque del primer tamaño a partir de los uno o más patrones de partición almacenados, basándose en la lista de patrones de partición, en donde el patrón de partición en la lista de patrones de partición se identifica mediante un índice de la lista de patrones de partición; determinar un patrón de partición para un bloque de profundidad de un segundo tamaño basándose en el patrón de partición determinado asociado al bloque del primer tamaño, en donde el segundo tamaño es mayor que el primer tamaño, en donde el bloque del primer tamaño está ubicado dentro del bloque de profundidad del segundo tamaño y en donde el bloque de profundidad es un bloque de un mapa de profundidad en un entorno de codificación tridimensional (3D) de vídeo; y codificar por intrapredicción el bloque de profundidad basándose en el patrón de partición determinado para el bloque de profundidad sin codificar por intrapredicción el bloque del primer tamaño utilizando el patrón de partición determinado asociado al bloque del primer tamaño, en donde el bloque del primer tamaño se codifica por intrapredicción como parte de la intrapredicción del bloque de profundidad.

[0012] En un ejemplo, la divulgación describe un medio de almacenamiento legible por ordenador que tiene instrucciones almacenadas en el mismo que, cuando son ejecutadas por uno o más procesadores de un dispositivo para la codificación de vídeo, hacen que los uno o más procesadores realicen el procedimiento de cualquiera de las reivindicaciones 1 a 4.

[0013] Los detalles de uno o más ejemplos de la divulgación se exponen en los dibujos adjuntos y en la descripción siguiente. Otras características, objetos y ventajas resultarán evidentes a partir de la descripción y de los dibujos, y a partir de las reivindicaciones.

La presente invención se expone en las reivindicaciones.

BREVE DESCRIPCIÓN DE LOS DIBUJOS**[0014]**

5 La figura 1 es un diagrama de bloques que ilustra un sistema ejemplar de codificación y decodificación de vídeo 10 que puede utilizar las técnicas de esta divulgación para la codificación de profundidad.

La figura 2 generalmente ilustra las direcciones de predicción asociadas a modalidades de intrapredicción direccionales.

10 Las figuras 3A y 3B son diagramas conceptuales que ilustran ejemplos de modalidades de modelado en profundidad (DMM).

15 Las figuras 4A a 4F son diagramas conceptuales que ilustran ejemplos de patrones de partición.

Las figuras 5A y 5B son diagramas conceptuales que ilustran la correlación de muestras de patrones de partición para una precisión de media muestra.

20 La figura 6 es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo que puede implementar técnicas para la codificación de profundidad.

La figura 7 es un diagrama de bloques que ilustra un ejemplo de un decodificador de vídeo que puede implementar técnicas para la codificación de profundidad.

25 La figura 8 es un diagrama de flujo que ilustra un procedimiento ejemplar de decodificación de datos de vídeo.

La figura 9 es un diagrama de flujo que ilustra un procedimiento ejemplar de codificación de datos de vídeo.

30 La figura 10 es un diagrama conceptual que ilustra una modalidad de codificación de cadenas de fronteras regionales.

DESCRIPCIÓN DETALLADA

35 **[0015]** En la codificación de vídeo tridimensional (3D), tal como de acuerdo a la norma 3D-HEVC (codificación de vídeo de alta eficacia), los datos de vídeo tridimensional se representan utilizando el vídeo de vista múltiple más el formato de profundidad, en el que los datos de vídeo capturados (también denominados componentes de textura de una vista de textura) están asociados a los correspondientes mapas de profundidad de una vista de profundidad. Por ejemplo, un componente de textura, de la vista de textura, representa el contenido de vídeo real, y un componente de profundidad, correspondiente al componente de textura, proporciona un mapa de profundidad que representa la profundidad relativa de los píxeles en el componente de textura. Un codificador de vídeo está configurado para codificar los componentes de textura y mapas de profundidad y multiplexar los componentes de textura y mapas de profundidad en un flujo de bits de vídeo tridimensional. Un decodificador de vídeo recibe el flujo de bits de vídeo tridimensional y decodifica los componentes de textura y los mapas de profundidad para reconstruir los datos de vídeo tridimensional.

45 **[0016]** Para aprovechar las capacidades de codificación de vídeo de un codificador de vídeo y las capacidades de decodificación de un decodificador de vídeo, los mapas de profundidad se forman como vídeo en escala de grises donde las muestras de luma representan los valores de profundidad. De esta manera, los valores de profundidad pueden codificarse y decodificarse utilizando las capacidades existentes del codificador de vídeo y del decodificador de vídeo, en lugar de tener que utilizar algunas técnicas especializadas adicionales de codificación y decodificación para codificar y decodificar los valores de profundidad.

50 **[0017]** Por ejemplo, el codificador de vídeo se puede configurar para utilizar la codificación por interpredicción y/o la codificación por intrapredicción y el decodificador de vídeo se puede configurar para utilizar la decodificación por interpredicción y/o la decodificación por intrapredicción para codificar imágenes de vídeo. El codificador de vídeo y el decodificador de vídeo, de manera similar, pueden usar, respectivamente, técnicas de codificación y decodificación por interpredicción e intrapredicción para codificar los mapas de profundidad.

55 **[0018]** En algunos casos, los mapas de profundidad incluyen bordes agudos y áreas constantes, donde se produce un borde agudo cuando hay una diferencia relativamente grande entre los valores de luma en un lado del borde y los valores de luma en el otro lado del borde. Debido a tales estadísticas diferentes de muestras de mapas de profundidad (por ejemplo, valores de luma), puede haber diferentes esquemas de codificación que están diseñados para mapas de profundidad basándose en un códec de vídeo bidimensional (2D). Por ejemplo, para la codificación de vídeo bidimensional, puede no haber mapas de profundidad. Sin embargo, para la codificación de vídeo tridimensional que incluye mapas de profundidad, las técnicas adicionales de codificación de vídeo pueden ser útiles para codificar y decodificar mapas de profundidad.

[0019] Por ejemplo, la norma 3D-HEVC extiende los conceptos para la codificación de vídeo bidimensional, tal como se define en la norma HEVC, a la codificación de vídeo tridimensional. La norma 3D-HEVC utiliza las modalidades de intrapredicción que se definen en la norma HEVC para la codificación y decodificación por intrapredicción. Además, la norma 3D-HEVC introdujo las modalidades de modelado de profundidad (DMM) junto con las modalidades de intrapredicción de la HEVC para codificar o decodificar una unidad de intrapredicción de un fragmento de un mapa de profundidad (es decir, intrapredicir una unidad de predicción (PU) de un fragmento de profundidad).

[0020] En las DMM, un bloque de un mapa de profundidad se divide en dos regiones, donde cada región está representada por un valor constante. El codificador de vídeo determina valores predictivos para cada una de las regiones que usa el codificador de vídeo para intrapredicir el bloque del mapa de profundidad. El codificador de vídeo también puede señalar los valores predictivos para cada una de las regiones al decodificador de vídeo, o el decodificador de vídeo puede configurarse para determinar los valores predictivos sin señalización explícita por parte del codificador de vídeo. En cualquier caso, el decodificador de vídeo puede utilizar los valores predictivos para intrapredicir el bloque del mapa de profundidad.

[0021] Una de las formas de dividir un bloque de un mapa de profundidad (denominado bloque de profundidad) se conoce como partición de minicuñas. En la partición de minicuñas, el codificador de vídeo determina una línea lineal que biseca el bloque de profundidad, creando así dos regiones. De esta manera, la partición de minicuñas se puede considerar como una división basada en líneas y, en algunos ejemplos, forman particiones no rectangulares (pero puede ser posible formar particiones rectangulares). La línea lineal puede comenzar desde un punto en un lado del bloque de profundidad y terminar en un punto en un lado opuesto o en el lado ortogonal del bloque de profundidad. Como ejemplo, en una orientación, la línea lineal puede comenzar desde un punto en un lado izquierdo del bloque de profundidad y terminar en un punto en un lado superior del bloque de profundidad. En otra orientación, la línea lineal puede comenzar desde un punto en un lado superior del bloque de profundidad y terminar en un punto en un lado inferior del bloque de profundidad.

[0022] Un patrón de minicuñas se refiere a una forma en que un bloque de un mapa de profundidad se puede dividir en dos regiones con una línea lineal bisectriz, y el número de patrones de minicuñas que pueden existir para un bloque puede ser una función del tamaño del bloque. Por ejemplo, para una resolución dada (por ejemplo, de píxel, de medio píxel o de cuarto de píxel), para bloques de menor tamaño (por ejemplo, bloques de 4x4), hay menos puntos a lo largo de los lados de los bloques en comparación con bloques de mayor tamaño (por ejemplo, 64x64). Por lo tanto, hay menos puntos de inicio y final a lo largo de cada lado de los bloques de tamaños más pequeños, en comparación con los bloques de tamaños más grandes, lo que da como resultado menos patrones de minicuñas.

[0023] En los ejemplos donde se utiliza una de las modalidades DMM, el codificador de vídeo puede determinar un patrón de minicuñas para el bloque de profundidad, e intrapredicir el patrón de minicuñas basándose en el patrón de minicuñas determinado. Debido a que el decodificador de vídeo se puede configurar para realizar el proceso inverso para decodificar el bloque, el decodificador de vídeo puede determinar el mismo patrón de minicuñas que determinó el codificador de vídeo. Por ejemplo, en algunos ejemplos, el codificador de vídeo y el decodificador de vídeo pueden almacenar, cada uno, una lista de patrones de minicuñas. El codificador de vídeo puede señalar en el flujo de bits de vídeo un índice de la lista de patrones de minicuñas, que identifica el patrón de minicuñas determinado. El decodificador de vídeo puede luego determinar el mismo patrón de minicuñas que el codificador de vídeo, basándose en el índice señalado en la lista de patrones de minicuñas. Como otro ejemplo, el codificador de vídeo puede determinar el patrón de minicuñas para el bloque de profundidad a partir de las características de contenido de vídeo del correspondiente bloque de textura. El decodificador de vídeo puede configurarse para implementar las mismas técnicas que el codificador de vídeo para determinar el patrón de minicuñas, de modo que el patrón de minicuñas que determine el decodificador de vídeo y el patrón de minicuñas que determine el codificador de vídeo sean el mismo patrón de minicuñas.

[0024] En los ejemplos anteriores para determinar el patrón de minicuñas, el codificador de vídeo y el decodificador de vídeo pueden configurarse para almacenar información para todos los patrones de minicuñas para todos los tamaños de bloque. Sin embargo, para bloques de mayor tamaño, la cantidad de memoria necesaria para almacenar el número relativamente grande de patrones de minicuñas puede ser indeseablemente grande. Por ejemplo, algunos borradores de la norma 3D-HEVC propusieron no tener ningún patrón de minicuñas para bloques de profundidad de tamaño 64x64. En otras palabras, se consideró que las desventajas de tener memoria de gran tamaño para almacenar los patrones de minicuñas para bloques de profundidad de tamaño 64x64 contrarrestaban los beneficios de usar los patrones de minicuñas para intrapredicir bloques de profundidad de tamaño 64x64.

[0025] En las técnicas descritas en esta divulgación, un codificador de vídeo (por ejemplo, el codificador de vídeo o el decodificador de vídeo) puede determinar un patrón de partición (por ejemplo, un patrón de minicuñas) a partir de patrones de partición de bloques de un primer tamaño (por ejemplo, determinar un patrón de partición asociado a bloques de un primer tamaño), y determinar un patrón de partición para un bloque de profundidad de un segundo tamaño, basándose en el patrón de partición determinado asociado a bloques del primer tamaño, donde el primer tamaño es más pequeño que el segundo tamaño. El codificador de vídeo puede codificar (por ejemplo, codificar o decodificar) por intrapredicción el bloque de profundidad del segundo tamaño basándose en el patrón de partición determinado. En algunos ejemplos, el codificador de vídeo puede utilizar un patrón de partición a partir de patrones

de partición de un subbloque (por ejemplo, un bloque dentro) del bloque de profundidad del segundo tamaño. Como ejemplo, el bloque de profundidad del segundo tamaño puede ser un bloque de tamaño $N \times N$, y el subbloque puede ser un bloque de tamaño $M \times M$, donde M es menor que N , y está dentro del bloque de tamaño $N \times N$.

5 **[0026]** En algunos ejemplos, la cantidad de patrones de partición que el codificador de vídeo requiere que sean almacenados para un tamaño de bloque particular puede reducirse si (es decir, basándose en que) un patrón de partición para un bloque de profundidad del tamaño de bloque particular se determina a partir de un patrón de partición asociado a bloques de menor tamaño. Por ejemplo, el codificador de vídeo puede almacenar los patrones de partición asociados al bloque de menor tamaño y luego determinar posteriormente un patrón de partición para el bloque de mayor tamaño a partir de los patrones de partición almacenados del bloque de menor tamaño.

15 **[0027]** Por ejemplo, supongamos que anteriormente un codificador de vídeo almacenaba un número Y de patrones de partición para un bloque de profundidad de un primer tamaño y un número X de patrones de partición para un bloque de profundidad de un segundo tamaño, donde el primer tamaño es más pequeño que el segundo. En este ejemplo, dado que el codificador de vídeo puede usar los Y patrones de partición para determinar el patrón de partición para un bloque de un segundo tamaño, es posible que el codificador de vídeo necesite almacenar menos del número X de patrones de partición para bloques de profundidad del segundo tamaño.

20 **[0028]** Puede ser posible que haya algunos patrones de partición asociados a un bloque del segundo tamaño que no se puedan determinar a partir de un patrón de partición para un bloque del primer tamaño. En consecuencia, en algunos ejemplos, el codificador de vídeo aún puede almacenar algunos (por ejemplo, al menos uno de los) patrones de partición para bloques de profundidad del segundo tamaño, pero el número de patrones de partición que deben almacenarse puede reducirse con respecto a las técnicas anteriores. En algunos ejemplos, el codificador de vídeo puede no almacenar ningún patrón de partición para un bloque del segundo tamaño, más grande, y en cambio puede determinar el patrón de partición para un bloque del segundo tamaño, más grande, del patrón de partición para un bloque del primer tamaño, más pequeño.

30 **[0029]** La figura 1 es un diagrama de bloques que ilustra un sistema ejemplar de codificación y decodificación de vídeo 10 que puede utilizar las técnicas de esta divulgación para la codificación de profundidad. Como se muestra en la figura 1, el sistema 10 incluye un dispositivo de origen 12 que proporciona datos de vídeo codificados, a decodificar posteriormente por un dispositivo de destino 14. En particular, el dispositivo de origen 12 proporciona los datos de vídeo al dispositivo de destino 14 mediante un medio legible por ordenador 16. El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender cualquiera entre una amplia gama de dispositivos, incluidos ordenadores de escritorio, ordenadores plegables (es decir, portátiles), ordenadores de tableta, decodificadores, equipos telefónicos de mano tales como los denominados teléfonos "inteligentes", los denominados paneles "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de transmisión continua de vídeo o similares. En algunos casos, el dispositivo de origen 12 y el dispositivo de destino 14 pueden estar equipados para la comunicación inalámbrica.

40 **[0030]** El dispositivo de destino 14 puede recibir los datos de vídeo codificados a decodificar mediante el medio legible por ordenador 16. El medio legible por ordenador 16 puede comprender cualquier tipo de medio o dispositivo capaz de transportar los datos de vídeo codificados desde el dispositivo de origen 12 al dispositivo de destino 14. En un ejemplo, el medio legible por ordenador 16 puede comprender un medio de comunicación para habilitar el dispositivo de origen 12 para transmitir datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real.

50 **[0031]** Los datos de vídeo codificados se pueden modular de acuerdo a una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitir al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación, inalámbrica o cableada, tal como un espectro de radiofrecuencia (RF) o una o más líneas físicas de transmisión. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área amplia o una red global tal como Internet. El medio de comunicación puede incluir encaminadores, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo de origen 12 hasta el dispositivo de destino 14.

55 **[0032]** En algunos ejemplos, pueden emitirse datos codificados desde la interfaz de salida 22 hasta un dispositivo de almacenamiento. De forma similar, se puede acceder a los datos codificados desde el dispositivo de almacenamiento mediante una interfaz de entrada. El dispositivo de almacenamiento puede incluir cualquiera entre varios medios de almacenamiento de datos, distribuidos o de acceso local, tales como un disco duro, discos Blu-ray, DVD, CD-ROM, memoria flash, memoria volátil o no volátil, o cualquier otro medio de almacenamiento digital adecuado para almacenar datos de vídeo codificados. En un ejemplo adicional, el dispositivo de almacenamiento puede corresponder a un servidor de ficheros o a otro dispositivo de almacenamiento intermedio que pueda almacenar el vídeo codificado generado por el dispositivo de origen 12.

65 **[0033]** El dispositivo de destino 14 puede acceder a datos de vídeo almacenados desde el dispositivo de almacenamiento, mediante transmisión continua o descarga. El servidor de ficheros puede ser cualquier tipo de servidor capaz de almacenar datos de vídeo codificados y transmitir esos datos de vídeo codificados al dispositivo de

destino 14. Los servidores ejemplares de ficheros incluyen un servidor de la Red (por ejemplo, para una sede de la Red), un servidor del FTP, dispositivos de almacenamiento conectados en red (NAS) o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de cualquier conexión de datos estándar, incluida una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión de Wi-Fi), una conexión cableada (por ejemplo, DSL, módem de cable, etc.) o una combinación de ambas que sea adecuada para acceder a datos de vídeo codificados almacenados en un servidor de ficheros. La transmisión de datos de vídeo codificados desde el dispositivo de almacenamiento puede ser una transmisión continua, una transmisión de descarga o una combinación de ambas.

[0034] Las técnicas de esta divulgación no están limitadas necesariamente a aplicaciones o configuraciones inalámbricas. Las técnicas se pueden aplicar a la codificación de vídeo como soporte de cualquiera entre varias aplicaciones de multimedios, tales como difusiones de televisión por el aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones continuas de vídeo por Internet, tales como la transmisión continua adaptativa dinámica por HTTP (DASH), vídeo digital que se codifica en un medio de almacenamiento de datos, decodificación de vídeo digital almacenado en un medio de almacenamiento de datos u otras aplicaciones. En algunos ejemplos, el sistema 10 puede estar configurado para prestar soporte a una transmisión de vídeo unidireccional o bidireccional, a fin de prestar soporte a aplicaciones tales como la transmisión de vídeo en tiempo real, la reproducción de vídeo, la difusión de vídeo y/o la videotelefonía.

[0035] En el ejemplo de la figura 1, el dispositivo de origen 12 incluye un origen de vídeo 18, un codificador de vídeo 20 y una interfaz de salida 22. El dispositivo de destino 14 incluye una interfaz de entrada 28, un decodificador de vídeo 30 y un dispositivo de visualización 32. De acuerdo a esta divulgación, el codificador de vídeo 20 del dispositivo de origen 12 puede estar configurado para aplicar las técnicas de predicción de vectores de movimiento en la codificación de múltiples visualizaciones. En otros ejemplos, un dispositivo de origen y un dispositivo de destino pueden incluir otros componentes o disposiciones. Por ejemplo, el dispositivo de origen 12 puede recibir datos de vídeo desde un origen de vídeo externo 18, tal como una cámara externa. Del mismo modo, el dispositivo de destino 14 puede interactuar con un dispositivo de visualización externo, en lugar de incluir un dispositivo de visualización integrado.

[0036] El sistema ilustrado 10 de la figura 1 es simplemente un ejemplo. Las técnicas de acuerdo a esta divulgación pueden realizarse mediante cualquier dispositivo de codificación y/o decodificación de vídeo digital. Aunque, en general, las técnicas de esta divulgación son llevadas a cabo por un dispositivo de codificación de vídeo y un dispositivo de decodificación de vídeo, las técnicas también pueden ser llevadas a cabo por un codificador/decodificador de vídeo, denominado habitualmente "CÓDEC". Además, las técnicas de esta divulgación también se pueden llevar a cabo mediante un preprocesador de vídeo. El dispositivo de origen 12 y el dispositivo de destino 14 son simplemente ejemplos de dichos dispositivos de codificación en los que el dispositivo de origen 12 genera datos de vídeo codificados para su transmisión al dispositivo de destino 14. En algunos ejemplos, los dispositivos 12, 14 pueden funcionar de manera esencialmente simétrica, de modo que cada uno de los dispositivos 12, 14 incluya componentes de codificación y decodificación de vídeo. De ahí que el sistema 10 pueda prestar soporte a la transmisión de vídeo unidireccional o bidireccional entre los dispositivos de vídeo 12, 14, por ejemplo, para la transmisión continua de vídeo, la reproducción de vídeo, la difusión de vídeo o la videotelefonía.

[0037] El origen de vídeo 18 del dispositivo de origen 12 puede incluir un dispositivo de captura de vídeo, tal como una cámara de vídeo, un archivo de vídeo que contiene vídeo capturado previamente y/o una interfaz de suministro de vídeo para recibir vídeo desde un proveedor de contenidos de vídeo. Como otra alternativa, el origen de vídeo 18 puede generar datos basados en gráficos de ordenador como el vídeo de origen, o una combinación de vídeo en directo, vídeo archivado y vídeo generado por ordenador. En algunos casos, si el origen de vídeo 18 es una videocámara, el dispositivo de origen 12 y el dispositivo de destino 14 pueden formar los llamados teléfonos con cámara o videoteléfonos. Sin embargo, como se ha mencionado anteriormente, las técnicas descritas en esta divulgación pueden ser aplicables a la codificación de vídeo en general, y se pueden aplicar a aplicaciones inalámbricas y/o cableadas. En cada caso, el codificador de vídeo 20 puede codificar el vídeo capturado, precapturado o generado por ordenador. Después, la información de vídeo codificada puede ser proporcionada por la interfaz de salida 22 a un medio legible por ordenador 16.

[0038] El medio legible por ordenador 16 puede incluir medios transitorios, tales como una difusión inalámbrica o una transmisión de red cableada, o medios de almacenamiento (es decir, medios de almacenamiento no transitorios), tales como un disco duro, una unidad de memoria flash, un disco compacto, un disco de vídeo digital, un disco Blu-ray u otros medios legibles por ordenador. En algunos ejemplos, un servidor de red (no mostrado) puede recibir datos de vídeo codificados desde el dispositivo de origen 12 y proporcionar los datos de vídeo codificados al dispositivo de destino 14, por ejemplo, mediante transmisión por red. De forma similar, un dispositivo informático de una utilidad de producción de un medio, tal como una utilidad de grabación de discos, puede recibir datos de vídeo codificados desde el dispositivo de origen 12 y producir un disco que contiene los datos de vídeo codificados. Por lo tanto, se puede entender que el medio legible por ordenador 16 incluye uno o más medios legibles por ordenador de diversas formas, en varios ejemplos.

[0039] Esta divulgación puede referirse en general al codificador de vídeo 20 que "señaliza" cierta información a otro

dispositivo, tal como el decodificador de vídeo 30. Sin embargo, debería entenderse que el codificador de vídeo 20 puede señalar información asociando ciertos elementos sintácticos a varias partes codificadas de datos de vídeo. Es decir, el codificador de vídeo 20 puede "señalizar" datos almacenando ciertos elementos sintácticos en cabeceras de varias partes codificadas de datos de vídeo. En algunos casos, dichos elementos sintácticos pueden ser codificados y almacenados (por ejemplo, almacenados en el medio legible por ordenador 16) antes de ser recibidos y decodificados por el decodificador de vídeo 30. Por lo tanto, el término "señalización" puede referirse, en general, a la comunicación de datos sintácticos u otros, usados para decodificar datos de vídeo comprimidos, ya sea que dicha comunicación se produzca en tiempo real o casi real o durante un intervalo de tiempo, tal como podría ocurrir cuando se almacenan elementos sintácticos en un medio en el momento de la codificación, que luego pueden ser recuperados por un dispositivo de decodificación en cualquier momento después de haber sido almacenados en este medio.

[0040] La interfaz de entrada 28 del dispositivo de destino 14 recibe información desde el medio legible por ordenador 16. La información del medio legible por ordenador 16 puede incluir información sintáctica definida por el codificador de vídeo 20, que también es usada por el decodificador de vídeo 30, que incluye elementos sintácticos que describen características y/o procesamiento de bloques y otras unidades codificadas, por ejemplo, los GOP. El dispositivo de visualización 32 muestra los datos de vídeo decodificados a un usuario, y puede comprender cualquiera entre varios 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.

[0041] Aunque no se muestra en la figura 1, en algunos aspectos, tanto el codificador de vídeo 20 como el decodificador de vídeo 30 pueden estar integrados con un codificador y decodificador de audio, y pueden incluir unidades MUX-DEMUX adecuadas, u otro tipo de hardware y software, para gestionar la codificación tanto de audio como de vídeo en un flujo de datos común o en flujos de datos independientes. Si procede, las unidades MUX-DEMUX pueden ser conformes al protocolo multiplexador ITU H.223 o a otros protocolos tales como el protocolo de datagramas de usuario (UDP).

[0042] Tanto el codificador de vídeo 20 como el decodificador de vídeo 30 puede implementarse como cualquiera entre varios circuitos codificadores o decodificadores adecuados, según corresponda, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), formaciones de compuertas programables in situ (FPGA), circuitos de lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. Tanto el codificador de vídeo 20 como el decodificador de vídeo 30 pueden estar incluidos en uno o más codificadores o decodificadores, cada uno de los cuales puede estar integrado como parte de un codificador/decodificador (CÓDEC) de vídeo combinado. Un dispositivo que incluye el codificador de vídeo 20 y/o el decodificador de vídeo 30 puede comprender un circuito integrado, un microprocesador y/o un dispositivo de comunicación inalámbrica, tal como un teléfono celular.

[0043] Un ejemplo de una norma de codificación de vídeo incluye la norma ITU-T H.264/MPEG-4 (AVC), que fue formulada por el Grupo de Expertos en Codificación de Vídeo (VCEG) de la ITU-T junto con el Grupo de Expertos en Imágenes en Movimiento (MPEG) de la ISO/IEC, como el producto de una asociación colectiva conocida como el Equipo Conjunto de Vídeo (JVT). Otra norma de codificación de vídeo incluye la norma H.264, que incluye sus extensiones de Codificación de Vídeo Ajustable a Escala (SVC) y de Codificación de Vídeo de Vistas Múltiples (MVC). La norma H.264 se describe en la Recomendación H.264 de la ITU-T, Codificación Avanzada de Vídeo para servicios audiovisuales genéricos, por el Grupo de Estudio de la ITU-T. El equipo de vídeo conjunto (JVT) continúa trabajando en extensiones para la H.264/MPEG-4 AVC. El más reciente borrador conjunto de la MVC se describe en "Advanced video coding for generic audiovisual services [Codificación de vídeo avanzada para servicios audiovisuales genéricos]", Recomendación H.264 de la ITU-T, marzo de 2010.

[0044] Como alternativa, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar de acuerdo a una norma de Codificación de Vídeo de Alta Eficacia (HEVC) y a extensiones de la norma HEVC, y pueden ser conformes al Modelo de Prueba de la HEVC (HM). La HEVC fue desarrollada por el Equipo de Colaboración Conjunta sobre Codificación de Vídeo (JCT-VC) del Grupo de Expertos en Codificación de Vídeo (VCEG) de la ITU-T y el Grupo de Expertos en Imágenes en Movimiento (MPEG) de la ISO/IEC. Un borrador reciente de la HEVC está disponible en http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v14.zip. Los esfuerzos de normalización de la HEVC se basaron en un modelo en evolución de un dispositivo de codificación de vídeo denominado modelo de prueba de la HEVC (HM). El HM supone varias capacidades adicionales de los dispositivos de codificación de vídeo respecto a los dispositivos existentes, de acuerdo con, por ejemplo, la norma ITU-T H.264/AVC. Por ejemplo, mientras que la norma H.264 proporciona nueve modalidades de codificación por intrapredicción, el HM puede proporcionar hasta treinta y cinco modalidades de codificación por intrapredicción.

[0045] En general, el modelo de funcionamiento del HM describe que una imagen (o "trama") de vídeo puede dividirse en una secuencia de bloques arbolados o unidades de codificación de máximo tamaño (LCU), que incluyen muestras tanto de luma como de croma. Los datos sintácticos dentro de un flujo de bits pueden definir un tamaño para la LCU, que es la mayor unidad de codificación en lo que respecta al número de píxeles. Un fragmento incluye un número de bloques arbolados consecutivos en orden de codificación. Una imagen puede dividirse en uno o más fragmentos. Cada bloque arbolado puede dividirse en unidades de codificación (CU) de acuerdo a un árbol cuádruple.

En general, una estructura de datos de árbol cuádruple incluye un nodo por CU, con un nodo raíz correspondiente al bloque arbolado. 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.

5 **[0046]** Cada nodo de la estructura de datos de 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 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. Si una CU no está dividida
10 adicionalmente, se denomina CU hoja. En esta divulgación, cuatro sub-CU de una CU hoja también se denominarán CU hojas, incluso aunque no haya ninguna división explícita de la CU hoja original. Por ejemplo, si una CU de tamaño 16x16 no se divide adicionalmente, las cuatro sub-CU de tamaño 8x8 también se denominarán CU hojas aunque la CU de tamaño 16x16 no se haya dividido nunca.

15 **[0047]** Una CU tiene un propósito similar a un macrobloque de la norma H.264, excepto porque una CU no tiene una distinción de tamaño. Por ejemplo, un bloque arbolado puede dividirse en cuatro nodos hijos (también denominados sub-CU) y cada nodo hijo puede a su vez ser un nodo padre y dividirse en otros cuatro nodos hijos. Un nodo hijo final, no dividido, denominado nodo hoja del árbol cuádruple, comprende un nodo de codificación, también denominado CU hoja. Los datos sintácticos asociados a un flujo de bits codificado pueden definir un número máximo de veces en que puede dividirse un bloque arbolado, lo que se denomina profundidad de CU máxima, y también pueden definir un
20 tamaño mínimo de los nodos de codificación. Por consiguiente, un flujo de bits también puede definir la unidad de codificación más pequeña (SCU). Esta divulgación utiliza el término "bloque" para referirse a cualquiera entre una CU, PU o TU, en el contexto de la HEVC, o a estructuras de datos similares en el contexto de otras normas (por ejemplo, macrobloques y subbloques de los mismos en la norma H.264/AVC).

25 **[0048]** Una CU incluye un nodo de codificación y unidades de predicción (PU) y unidades de transformación (TU) asociadas al nodo de codificación. Un tamaño de la CU corresponde a un tamaño del nodo de codificación y debe ser de forma cuadrada. El tamaño de la CU puede variar desde 8x8 píxeles hasta el tamaño del bloque arbolado, con un máximo de 64x64 píxeles o más. Cada CU puede contener una o más PU y una o más TU. Los datos sintácticos asociados a una CU pueden describir, por ejemplo, la división de la CU en una o más PU. Las modalidades de división
30 pueden diferir entre sí si la CU está codificada en modalidad de salto o directa, codificada en modalidad de intrapredicción o codificada en modo de interpredicción. Las PU pueden dividirse para tener forma no cuadrada. Los datos sintácticos asociados a una CU también pueden describir, por ejemplo, la división de la CU en una o más TU de acuerdo con un árbol cuádruple. Una TU puede tener forma cuadrada o no cuadrada (por ejemplo, rectangular).

35 **[0049]** La norma HEVC admite transformaciones de acuerdo a las TU, que pueden ser diferentes para diferentes CU. El tamaño de las TU se basa típicamente en el tamaño de las PU dentro de una CU definida dada para una LCU dividida, aunque puede que no sea siempre así. Las TU son típicamente del mismo tamaño o de un tamaño más pequeño que las PU. En algunos ejemplos, las muestras residuales correspondientes a una CU pueden subdividirse en unidades más pequeñas utilizando una estructura de árbol cuádruple conocida como "árbol cuádruple residual"
40 (RQT). Los nodos hojas del RQT pueden denominarse unidades de transformación (TU). Los valores de diferencias de píxeles asociados a las TU pueden transformarse para generar coeficientes de transformación, que pueden cuantizarse.

45 **[0050]** Una CU hoja puede incluir una o más unidades de predicción (PU). En general, una PU representa una zona espacial correspondiente a la totalidad, o a una parte, de la CU correspondiente, y puede incluir datos para recuperar una muestra de referencia para la PU. Además, una PU incluye datos relacionados con la predicción. Por ejemplo, cuando la PU está codificada por intramodalidad (por ejemplo, codificada por intrapredicción) pueden incluirse datos para la PU en un árbol cuádruple residual (RQT), que puede incluir datos que describen una modalidad de intrapredicción para una TU correspondiente a la PU. Como otro ejemplo, cuando la PU está codificada por intermodalidad (por ejemplo, codificada por interpredicción), la PU puede incluir datos que definen uno o más vectores de movimiento para la PU. Los datos que definen el vector de movimiento para una PU 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, precisión de un cuarto de píxel o precisión de un octavo de píxel), una imagen de referencia a la que apunta el vector de movimiento y/o una lista de imágenes de referencia (por ejemplo, ListImgRef0 o ListImgRef1) para el vector de movimiento.
55

[0051] Una CU hoja que tiene una o más PU también puede incluir una o más unidades de transformación (TU). Las unidades de transformación pueden especificarse usando un RQT (también denominado estructura de árbol cuádruple de TU), como se ha expuesto anteriormente. Por ejemplo, un indicador de división puede indicar si una CU hoja está dividida en cuatro unidades de transformación. A continuación, cada unidad de transformación puede dividirse adicionalmente en sub-TU adicionales. Cuando una TU no se divide adicionalmente, puede denominarse una TU hoja. En general, en lo que respecta a la intracodificación, todas las TU hojas que pertenecen a una CU hoja comparten la misma modalidad de intrapredicción. Es decir, la misma modalidad de intrapredicción se aplica en general para calcular valores predichos para todas las TU de una CU hoja. En lo que respecta a la intracodificación, un codificador de vídeo
60 20 puede calcular un valor residual para cada TU hoja usando la modalidad de intrapredicción, como una diferencia entre la parte de la CU correspondiente a la TU y el bloque original. Una TU no está necesariamente limitada al tamaño

de una PU. De este modo, las TU pueden ser mayores o menores que una PU. En lo que respecta a la intracodificación (por ejemplo, la codificación por intrapredicción), una PU puede estar cosituada con una TU hoja correspondiente para la misma CU. En algunos ejemplos, el tamaño máximo de una TU hoja puede corresponder al tamaño de la CU hoja correspondiente.

5 **[0052]** Además, las TU de las CU hojas también pueden asociarse a respectivas estructuras de datos de árbol cuádruple, denominadas árboles cuádruples residuales (RQT). Es decir, una CU hoja puede incluir un árbol cuádruple que indica cómo la CU hoja está dividida en las TU. El nodo raíz de un árbol cuádruple de TU corresponde en general a una CU hoja, mientras que el nodo raíz de un árbol cuádruple de CU corresponde en general a un bloque arbolado (o LCU). Las TU del RQT que no están divididas se denominan TU hojas. En general, esta divulgación usa los términos CU y TU para hacer referencia a una CU hoja y a una TU hoja, respectivamente, a no ser que se indique lo contrario.

10 **[0053]** Una secuencia de vídeo incluye habitualmente una serie de imágenes. Según lo descrito en el presente documento, los términos "imagen" y "trama" pueden usarse de forma intercambiable. Es decir, la imagen que contiene datos de vídeo puede denominarse trama de vídeo o simplemente "trama". Un grupo de imágenes (GOP) comprende, en general, una serie de una o más de las imágenes de vídeo. Un GOP puede incluir datos sintácticos en una cabecera del GOP, una cabecera de una o más de las imágenes, o en otras ubicaciones, que describen una serie de imágenes incluidas en el GOP. Cada fragmento de una imagen puede incluir datos sintácticos de fragmento que describen una modalidad de codificación para el fragmento respectivo. El codificador de vídeo 20 actúa habitualmente sobre bloques de vídeo dentro de fragmentos de vídeo individuales con el fin de codificar los datos de vídeo. Un bloque de vídeo puede corresponder a un nodo de codificación dentro de una CU. Los bloques de vídeo pueden tener tamaños fijos o variables y pueden diferir en tamaño de acuerdo a una norma de codificación especificada.

15 **[0054]** En un ejemplo, el HM presta soporte a la predicción en diversos tamaños de PU. Si se supone que el tamaño de una CU particular es $2N \times 2N$, el HM da soporte a la intrapredicción en tamaños de PU de $2N \times 2N$ o $N \times N$, y a la interpredicción en tamaños de PU simétricas de $2N \times 2N$, $2N \times N$, $N \times 2N$ o $N \times N$. El HM también da soporte a la división asimétrica para la interpredicción en tamaños de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$ y $nR \times 2N$. En la división asimétrica, una dirección de una CU no está dividida, mientras que la otra dirección está dividida entre un 25 % y un 75 %. La parte de la CU correspondiente a la división del 25 % está indicada por una "n" seguida por una indicación de "arriba", "abajo", "izquierda" o "derecha". Así pues, por ejemplo, "2NxnU" se refiere a una CU de tamaño $2N \times 2N$ que está dividida horizontalmente, con una PU de tamaño $2N \times 0,5N$ encima y una PU de tamaño $2N \times 1,5N$ debajo.

20 **[0055]** En la presente divulgación, "NxN" y "N por N" pueden usarse indistintamente para hacer referencia a las dimensiones de píxeles de un bloque de vídeo en términos de dimensiones verticales y horizontales, por ejemplo, 16×16 píxeles o 16 por 16 píxeles. En general, un bloque de 16×16 tendrá 16 píxeles en una dirección vertical ($y = 16$) y 16 píxeles en una dirección horizontal ($x = 16$). Asimismo, un bloque de tamaño NxN tiene, en general, N píxeles en una dirección vertical y N píxeles en una dirección horizontal, donde N representa un valor entero no negativo. Los píxeles de un bloque se pueden disponer en filas y columnas. Además, no es necesario que los bloques tengan necesariamente el mismo número de píxeles en la dirección horizontal que en la dirección vertical. Por ejemplo, los bloques pueden comprender $N \times M$ píxeles, donde M no es necesariamente igual a N.

25 **[0056]** Tras la codificación intrapredictiva o interpredictiva, usando las PU de una CU, el codificador de vídeo 20 puede calcular datos residuales para las TU de la CU. Las PU pueden comprender datos sintácticos que describen un procedimiento o modalidad de generación de datos de píxeles predictivos en el dominio espacial (también denominado el dominio de píxeles) y las TU pueden comprender coeficientes en el dominio de transformación, tras la aplicación de una transformación, por ejemplo, una transformación de coseno discreta (DCT), una transformación entera, una transformación de ondículas o una transformación conceptualmente similar, en los datos de vídeo residuales. Los datos residuales pueden corresponder a diferencias de píxeles entre píxeles de la imagen no codificada y valores de predicción correspondientes a las PU. El codificador de vídeo 20 puede formar las TU incluyendo los datos residuales para la CU y, a continuación, transformar las TU para generar coeficientes de transformación para la CU.

30 **[0057]** Tras cualquier transformación para generar coeficientes de transformación, el codificador de vídeo 20 puede realizar la cuantización de los coeficientes de transformación. La cuantización se refiere, en general, a un proceso en el que unos coeficientes de transformación se cuantizan para reducir posiblemente la cantidad de datos usados para representar los coeficientes, proporcionando compresión adicional. El proceso de cuantización puede reducir la profundidad de bits asociada a algunos, o a la totalidad, de los coeficientes. Por ejemplo, un valor de n bits puede redondearse a la baja hasta un valor de m bits durante la cuantización, donde n es mayor que m .

35 **[0058]** Después de la cuantización, el codificador de vídeo 20 puede recorrer los coeficientes de transformación, produciendo un vector unidimensional a partir de la matriz bidimensional que incluye los coeficientes de transformación cuantizados. La exploración puede estar diseñada para colocar los coeficientes de mayor energía (y, por lo tanto, de menor frecuencia) al frente de la formación y para colocar los coeficientes de menor energía (y, por lo tanto, de mayor frecuencia) al final de la formación.

40 **[0059]** En algunos ejemplos, el codificador de vídeo 20 puede utilizar un orden de exploración predefinido para explorar los coeficientes de transformación cuantizados, para producir un vector serializado que se pueda codificar

por entropía. En otros ejemplos, el codificador de vídeo 20 puede realizar una exploración adaptativa. Después de explorar los coeficientes de transformación cuantizados para formar un vector unidimensional, el codificador de vídeo 20 puede realizar la codificación por entropía del vector unidimensional, por ejemplo, de acuerdo a la codificación de longitud variable adaptativa al contexto (CAVLC), la codificación aritmética binaria adaptativa al contexto (CABAC), la codificación aritmética binaria adaptativa al contexto basada en la sintaxis (SBAC), la codificación por entropía por división en intervalos de probabilidad (PIPE) o con otra metodología de codificación por entropía. El codificador de vídeo 20 también puede realizar la codificación por entropía de los elementos sintácticos asociados a los datos de vídeo codificados, para su uso por el decodificador de vídeo 30 en la decodificación de los datos de vídeo.

[0060] Además, el codificador de vídeo 20 puede enviar datos sintácticos, tales como datos sintácticos basados en bloques, datos sintácticos basados en imágenes y datos sintácticos basados en GOP, al decodificador de vídeo 30, por ejemplo, en una cabecera de imagen, una cabecera de bloque, una cabecera de fragmento o una cabecera de GOP. Los datos sintácticos del GOP pueden describir una serie de imágenes en el GOP respectivo, y los datos sintácticos de imagen pueden indicar una modalidad de codificación/predicción usada para codificar la imagen correspondiente.

[0061] El decodificador de vídeo 30 puede configurarse para realizar generalmente el procedimiento recíproco para decodificar los datos de vídeo y reconstruir las imágenes que el codificador de vídeo 20 utilizó con fines de codificación. Por ejemplo, el decodificador de vídeo 30 puede recibir los elementos de sintaxis y los datos de vídeo desde el flujo de bits señalizado y realizar operaciones recíprocas para la decodificación por intrapredicción y/o la decodificación por interpredicción de los datos de vídeo para reconstruir las imágenes.

[0062] La descripción anterior proporciona formas ejemplares en las que el codificador de vídeo 20 y el decodificador de vídeo 30 pueden codificar y decodificar datos de vídeo basándose en la norma HEVC. En las técnicas descritas en esta divulgación, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para la codificación y decodificación de vídeo tridimensional (3D). Por ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para la codificación y decodificación de vídeo tridimensional, utilizando normas de codificación de vídeo en desarrollo que aprovechan la norma de codificación de vídeo HEVC. Sin embargo, las técnicas descritas en esta divulgación no están limitadas de ese modo y pueden extenderse a otras técnicas de codificación y decodificación de vídeo tridimensional.

[0063] En el JCT-3V, hay dos extensiones de la HEVC que se están desarrollando, denominadas extensión de múltiples vistas (MV-HEVC) y extensión de vídeo tridimensional (3D-HEVC). El software de referencia 3D-HTM más reciente, versión 8.0 para la 3D-HEVC, se puede descargar, a partir del 13 de octubre de 2014, desde el siguiente enlace: [3D-HTM versión 8,0]: https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-8.0/. El más reciente borrador de trabajo (número de documento: E1001 (JCT3V-E1001)) está disponible en: http://phenix.it-sudparis.eu/jct2/doc_end_user/documents/5_Vienna/wg11/JCT3V-E1001-v3.zip (sin embargo, este enlace puede quedar inhabilitado). El documento JCT3V-E1001 se titula "3D-HEVC Draft Text 1 [Texto borrador 1 de 3D-HEVC]", de Sullivan et al., y también está disponible, a partir del 13 de octubre de 2014, en http://phenix.it-sudparis.eu/jct2/doc_end_user/current_document.php?id=1361.

[0064] Es decir, los datos de vídeo codificados usando técnicas de codificación de vídeo tridimensional pueden representarse y mostrarse para producir un efecto tridimensional. Por ejemplo, dos imágenes de diferentes vistas (es decir, correspondientes a dos perspectivas de cámara que tienen posiciones horizontales ligeramente diferentes) pueden mostrarse de manera esencialmente simultánea, de modo que una imagen sea vista por el ojo izquierdo de un espectador, y la otra imagen sea vista por el ojo derecho del espectador.

[0065] Este efecto tridimensional se puede lograr usando, por ejemplo, pantallas estereoscópicas o pantallas auto-estereoscópicas. Las pantallas estereoscópicas pueden usarse junto con gafas que filtran las dos imágenes debidamente. Por ejemplo, las gafas pasivas pueden filtrar las imágenes utilizando lentes polarizadas o lentes de diferentes colores para garantizar que el ojo adecuado vea la imagen adecuada. Las gafas activas, como otro ejemplo, pueden obturar rápidamente lentes alternas en coordinación con la pantalla estereoscópica, que puede alternar entre mostrar la imagen del ojo izquierdo y la del ojo derecho. Las pantallas auto-estereoscópicas muestran las dos imágenes de tal manera que no se necesiten gafas. Por ejemplo, las pantallas auto-estereoscópicas pueden incluir espejos o prismas que están configurados para hacer que cada imagen se proyecte en los ojos adecuados de un espectador.

[0066] En la codificación de vídeo tridimensional, existe una pluralidad de vistas y cada vista incluye una pluralidad de imágenes denominadas imágenes de textura e imágenes de profundidad o, simplemente, mapa de componentes de textura y de profundidad. Cada componente de textura puede corresponder a un mapa de profundidad. El componente de textura puede incluir el contenido de la imagen y el correspondiente mapa de profundidad indica la profundidad relativa de los píxeles en la textura. Los componentes de textura de diferentes vistas que han de mostrarse de forma esencialmente simultánea incluyen contenido de imagen similar, pero existe una disparidad horizontal entre los objetos en las texturas de las diferentes vistas. El mapa de textura y profundidad se describe con más detalle a continuación.

[0067] En la norma 3D-HEVC, una unidad de acceso incluye las imágenes de textura y sus correspondientes imágenes de profundidad, que se han de mostrar de forma esencialmente simultánea. Las imágenes de textura y las imágenes de profundidad en cada vista tienen un identificador de vista único (identificador de vista) o un índice de orden de vista para identificar a qué vista pertenecen las imágenes. Sin embargo, una imagen de profundidad y una imagen de textura de la misma vista pueden tener diferentes identificadores de capa (identificadores de capa).

[0068] Las técnicas de la presente divulgación se refieren a la codificación de datos de vídeo tridimensional mediante la codificación de componentes de textura y mapas de profundidad. En general, el término "textura" se usa para describir valores de luminancia (brillo o "luma") de una imagen y valores de crominancia (color o "croma") de la imagen. En algunos ejemplos, una imagen de textura (es decir, una ilustración de textura) puede incluir un conjunto de datos de luminancia y dos conjuntos de datos de crominancia, para matices azules (Cb) y matices rojos (Cr). En ciertos formatos de croma, tales como 4:2:2 o 4:2:0, los datos de croma se someten a muestreo de frecuencia reducida con respecto a los datos de luma. Es decir, la resolución espacial de los píxeles de crominancia puede ser menor que la resolución espacial de los correspondientes píxeles de luminancia (por ejemplo, la mitad o un cuarto de la resolución de luminancia).

[0069] Los datos de profundidad generalmente describen valores de profundidad para los datos de textura correspondientes. Por ejemplo, una imagen de profundidad (por ejemplo, una ilustración de profundidad) puede incluir un conjunto de píxeles de profundidad, cada uno de los cuales describe la profundidad para los correspondientes datos de textura de la correspondiente imagen de textura. Los datos de profundidad pueden usarse para determinar la disparidad horizontal para los datos de textura correspondientes. Por lo tanto, un dispositivo que recibe los datos de textura y profundidad puede exhibir una primera imagen de textura para una vista (por ejemplo, una vista de ojo izquierdo) y usar los datos de profundidad para modificar la primera imagen de textura, para generar una segunda imagen de textura para la otra vista (por ejemplo, una vista de ojo derecho), desplazando los valores de píxeles de la primera imagen en los valores de disparidad horizontal, determinados basándose en los valores de profundidad. En general, la disparidad horizontal (o simplemente "disparidad") describe el desplazamiento espacial horizontal de un píxel en una primera vista con respecto a un píxel correspondiente en la vista derecha, donde los dos píxeles corresponden a la misma parte del mismo objeto, según se representa en las dos vistas.

[0070] En otros ejemplos adicionales, pueden definirse datos de profundidad para los píxeles en una dimensión z perpendicular al plano de imagen, de tal manera que una profundidad asociada a un píxel dado se defina en relación con un plano de disparidad cero, definido para la imagen. Dicha profundidad puede usarse para crear disparidad horizontal para exhibir el píxel, de manera que el píxel se exhiba de manera diferente para los ojos izquierdo y derecho, en función del valor de profundidad de la dimensión z del píxel con respecto al plano de disparidad cero. El plano de disparidad cero puede cambiar para diferentes partes de una secuencia de vídeo, y la magnitud de la profundidad relativa al plano de disparidad cero también puede cambiar. Los píxeles situados en el plano de disparidad cero pueden definirse de forma similar para los ojos izquierdo y derecho. Los píxeles situados delante del plano de disparidad cero pueden mostrarse en diferentes ubicaciones para el ojo izquierdo y el derecho (por ejemplo, con disparidad horizontal) a fin de crear una percepción de que el píxel parece salir de la imagen en la dirección z perpendicular al plano de imagen. Los píxeles situados detrás el plano de disparidad nula pueden mostrarse con una leve difuminación, para una leve percepción de profundidad, o pueden mostrarse en ubicaciones diferentes para el ojo izquierdo y el derecho (por ejemplo, con disparidad horizontal que es opuesta a la de los píxeles situados delante del plano de disparidad cero). También pueden usarse muchas otras técnicas para transmitir o definir datos de profundidad para una imagen.

[0071] Los datos de vídeo bidimensional generalmente se codifican como una secuencia de imágenes discretas, cada una de las cuales corresponde a una instancia temporal particular. Es decir, cada imagen tiene un tiempo de reproducción asociado con respecto a los tiempos de reproducción de otras imágenes en la secuencia. Estas imágenes pueden ser consideradas ilustraciones de textura o imágenes de textura. En la codificación de vídeo tridimensional basada en la profundidad, cada imagen de textura en una secuencia también puede corresponder a una imagen de profundidad (también mencionada como mapa de profundidad). Es decir, un mapa de profundidad correspondiente a una imagen de textura describe datos de profundidad para la correspondiente imagen de textura. Los datos de vídeo de múltiples vistas pueden incluir datos para varias vistas diferentes, donde cada vista puede incluir una secuencia respectiva de imágenes de textura y correspondientes imágenes de profundidad.

[0072] Como se ha señalado anteriormente, las imágenes pueden corresponder a una instancia temporal particular. Los datos de vídeo pueden representarse usando una secuencia de unidades de acceso, donde cada unidad de acceso incluye todos los datos correspondientes a una instancia temporal particular. Así, por ejemplo, para datos de vídeo de vistas múltiples más profundidad, las imágenes de textura de cada vista para una instancia temporal común, más los mapas de profundidad para cada una de las imágenes de textura, pueden incluirse todos dentro de una unidad de acceso particular. Una unidad de acceso puede incluir datos para un componente de textura, correspondiente a una imagen de textura, y un componente de profundidad, correspondiente a un mapa de profundidad.

[0073] De esta manera, los datos de vídeo tridimensional pueden representarse utilizando un formato de vídeo de vistas múltiples más profundidad, en el cual las vistas capturadas o generadas (componente de textura) están asociadas a los correspondientes mapas de profundidad. Además, en la codificación de vídeo tridimensional, los componentes de textura y los mapas de profundidad se pueden codificar y multiplexar en un flujo de bits de vídeo

tridimensional. Los mapas de profundidad se pueden codificar como imágenes en escala de grises, donde las muestras de "luma" (es decir, píxeles) de los mapas de profundidad representan valores de profundidad. En general, un bloque de datos de profundidad (un bloque de muestras de un mapa de profundidad) se puede denominar bloque de profundidad. Un valor de profundidad puede referirse a un valor de luminancia asociado a una muestra de profundidad.

[0074] En cualquier caso, se pueden aplicar procedimientos de intracodificación e intercodificación (por ejemplo, codificación y decodificación por intrapredicción y codificación y decodificación por interpredicción) para la codificación de mapas de profundidad. Por ejemplo, como se ha descrito anteriormente, aunque el mapa de profundidad indica valores de profundidad para la correspondiente imagen de textura, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden codificar y decodificar el mapa de profundidad utilizando técnicas de codificación de vídeo porque el mapa de profundidad se forma como una imagen en escala de grises, donde las muestras de luma del mapa de profundidad indican el valor de profundidad para los píxeles correspondientes en la correspondiente imagen de textura.

[0075] Los mapas de profundidad habitualmente incluyen bordes nítidos y áreas constantes, y los bordes en los mapas de profundidad usualmente presentan fuertes correlaciones con los correspondientes datos de textura de la correspondiente imagen de textura. Debido a las diferentes estadísticas y correlaciones entre la textura y la profundidad correspondiente, diferentes esquemas de codificación han sido, y continúan siendo, diseñados para mapas de profundidad basándose en un códec de vídeo bidimensional. Por ejemplo, puede haber esquemas adicionales de codificación de vídeo que aprovechen las diferentes estadísticas y correlaciones entre la textura y la profundidad correspondiente para la codificación de vídeo de los mapas de profundidad, además de las disponibles en la norma HEVC básica.

[0076] Como ejemplo, en la norma HEVC actual, las técnicas de intrapredicción para un componente de luma de cada unidad de predicción (PU) pueden utilizar 33 modalidades de predicción angular (indizadas de 2 a 34), una modalidad DC (indizada con 1) y una modalidad Plana (indizada con 0). La figura 2 generalmente ilustra las direcciones de predicción asociadas a modalidades de intrapredicción direccionales. Por ejemplo, como se ha indicado anteriormente, la norma HEVC puede incluir treinta y cinco modalidades de intrapredicción, incluyendo una modalidad plana (modalidad 0), una modalidad de DC (modalidad 1) y 33 modalidades de predicción direccional (modalidades 2 a 34). Con la modalidad plana, la predicción se realiza utilizando una función llamada "plana". Con la modalidad de DC (por ejemplo, para producir un valor de DC predicho), la predicción se puede realizar basándose en un promedio de valores de píxel dentro del bloque. Con una modalidad de predicción direccional, la predicción se realiza en función de los píxeles reconstruidos de un bloque vecino a lo largo de una dirección particular (según lo indicado por la modalidad). En general, el extremo de cola de las flechas mostradas en la figura 2 representa un píxel relativo de los píxeles vecinos a partir de los cuales se recupera un valor, mientras que la cabeza de las flechas representa la dirección en la que el valor recuperado se propaga para formar un bloque predictivo.

[0077] La norma 3D-HEVC utiliza la misma definición de las modalidades de intrapredicción que las de la norma HEVC. Además, la norma 3D-HEVC introdujo las Modalidades de modelado en profundidad (DMM) junto con las modalidades de intrapredicción de la HEVC para codificar o decodificar por intrapredicción un bloque de profundidad (por ejemplo, una unidad de predicción) de un fragmento de profundidad de un mapa de profundidad. Las DMM puede ser más adecuadas para las representaciones de bordes nítidos en los mapas de profundidad para la codificación por interpredicción (codificación o decodificación) de mapas de profundidad.

[0078] Algunas versiones anteriores de los borradores de trabajo de la norma 3D-HEVC proporcionan cuatro modalidades de las DMM: Modalidad 1 (señalización explícita de minicuñas), Modalidad 2 (partición de minicuñas intrapredichas), Modalidad 3 (partición de minicuñas entre componentes) y Modalidad 4 (partición de Contorno entre componentes). En las cuatro modalidades, un codificador de vídeo, tal como el codificador de vídeo 20 o el decodificador de vídeo 30, puede dividir un bloque de profundidad en dos regiones especificadas por un patrón de las DMM, donde cada región está representada por un valor constante. El patrón de las DMM puede ser explícitamente señalado (modalidad 1), predicho por bloques espacialmente vecinos (modalidad 2) o predicho usando un bloque de textura cosituado (modalidad 3 y modalidad 4).

[0079] Algunas versiones de los borradores de trabajo de la norma 3D-HEVC eliminaron la modalidad 2 de las DMM, dejando las modalidades 1, 3 y 4 de las DMM. Algunas versiones de los borradores de trabajo de la norma 3D-HEVC también eliminaron la modalidad 3 de las DMM, dejando las modalidades 1 y 4 de las DMM.

[0080] En otras palabras, hay nuevas modalidades de intrapredicción en las DMM. En estas modalidades, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para dividir un bloque de profundidad en dos regiones especificadas por un patrón de las DMM (denominado patrón de partición), donde cada región está representada por un valor constante. Por ejemplo, el codificador de vídeo 20 puede configurarse para usar diferentes patrones de partición (los ejemplos de los patrones de partición se describen en lo que antecede) para codificar por intrapredicción un bloque de profundidad, y determinar qué patrón de partición proporcionó la codificación óptima (por ejemplo, en términos de compresión y calidad de vídeo). El codificador de vídeo 20 puede entonces codificar por intrapredicción el bloque de profundidad usando el patrón de partición determinado. Debido a que el decodificador de vídeo 30 realiza el proceso recíproco al del codificador de vídeo 20 para decodificar por intrapredicción el bloque de profundidad, el decodificador de vídeo 30 puede configurarse para determinar el mismo patrón de partición que el

codificador de vídeo 20 determinó para decodificar por intrapredicción el bloque de profundidad.

- 5 **[0081]** El codificador de vídeo 20 puede señalar información que indica la modalidad de modelado en profundidad al decodificador de vídeo 30, y el decodificador de vídeo 30 puede determinar el patrón de partición a partir de la información señalizada que indica la modalidad de modelado en profundidad. Por ejemplo, si el codificador de vídeo 20 señala información que indica que la modalidad de modelado en profundidad es uno (modalidad 1 de las DMM), entonces el decodificador de vídeo 30 puede configurarse para analizar sintácticamente el flujo de bits para obtener información del codificador de vídeo 20 que identifica el patrón de partición para el bloque de profundidad. En otras palabras, si el codificador de vídeo 20 determina que se ha de usar la modalidad 1 de las DMM, el codificador de vídeo 10 20 puede señalar explícitamente información que indica que se usa una modalidad de las DMM y señalar información que usa el decodificador de vídeo 30 para identificar el patrón de partición para el bloque de profundidad. De esta manera, el patrón de partición (por ejemplo, el patrón de las DMM) que el codificador de vídeo 20 y el decodificador de vídeo 30 usan para la codificación por intrapredicción del bloque de profundidad es el mismo.
- 15 **[0082]** Para las modalidades 3 y 4 de las DMM, el codificador de vídeo 20 puede señalar información que indica que la modalidad de las DMM es la modalidad 3 o 4, pero puede no señalar información que identifique el patrón de partición para el bloque de profundidad. En cambio, el decodificador de vídeo 30 se puede configurar para determinar el patrón de partición para el bloque de profundidad a partir del bloque de textura cosituado en la correspondiente imagen de textura. El codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse, cada uno, para 20 implementar el mismo proceso para determinar el patrón de partición (por ejemplo, el patrón de las DMM) a partir del bloque de textura cosituado en la correspondiente imagen de textura para la modalidad 3 de las DMM, y configurarse para implementar el mismo proceso para determinar el patrón de partición (por ejemplo, el patrón de las DMM) a partir del bloque de textura cosituado en la correspondiente imagen de textura para la modalidad 4 de las DMM.
- 25 **[0083]** Hay dos tipos de modalidad de partición definidos en las DMM, incluidas la partición de minicuñas y la partición de Contorno. Las figuras 3A y 3B son diagramas conceptuales que ilustran ejemplos de modalidades de modelado en profundidad (DMM). La figura 3A ilustra un ejemplo de un patrón de minicuñas para un bloque de tamaño 8x8, y la figura 3B ilustra en el ejemplo de un patrón de Contorno para un bloque de tamaño 8x8.
- 30 **[0084]** La figura 3A, por ejemplo, ilustra el bloque de profundidad 110 que se divide usando la partición de minicuñas, y la figura 3B, como otro ejemplo, ilustra el bloque de profundidad 130 que se divide usando la partición de Contorno. La norma 3D-HEVC incluye técnicas para las modalidades de modelado de profundidad (DMM) para dividir bloques, junto con las modalidades de intrapredicción para codificar una unidad de intrapredicción de un fragmento de profundidad. La versión 3.1 de HTM aplica un procedimiento de las DMM para la intracodificación de mapas de 35 profundidad que, en algunos casos, puede representar mejor bordes más nítidos en mapas de profundidad.
- [0085]** Hay dos modelos de particiones definidos en las DMM, incluidas la partición de minicuñas y la partición de Contorno. De nuevo, la figura 3A ilustra un ejemplo de partición de minicuñas, y la figura 3B ilustra un ejemplo de 40 partición de Contorno. Cada cuadrado individual dentro de los bloques de profundidad 110 y 130 representa un respectivo píxel individual de los bloques de profundidad 110 y 130, respectivamente. Los valores numéricos dentro de los cuadrados representan si el píxel correspondiente pertenece a la región 112 (valor "0" en el ejemplo de la figura 3A) o a la región 114 (valor "1" en el ejemplo de la figura 3A). El sombreado también se usa en la figura 3A para indicar si un píxel pertenece a la región 112 (cuadrados blancos) o a la región 114 (cuadrados grises sombreados).
- 45 **[0086]** Cada patrón (es decir, tanto de minicuñas como de Contorno) se puede definir mediante una formación de tamaño $u_B \times v_B$ de dígitos binarios que etiquetan si la muestra correspondiente (es decir, el píxel) pertenece a la región P_1 o P_2 (donde P_1 corresponde a la región 112 en la figura 3A y a la región 132 en la figura 3B, y P_2 corresponde a la 50 región 114 en la figura 3A y a la región 134A, 134B en la figura 3B), donde u_B y v_B representan, respectivamente, el tamaño horizontal y vertical de la PU actual. En los ejemplos de la figura 3A y la figura 3B, la PU corresponde, respectivamente, a los bloques 110 y 130. Los codificadores de vídeo, tales como el codificador de vídeo 20 y el decodificador de vídeo 30, pueden inicializar patrones de minicuñas al comienzo de la codificación (por ejemplo, el comienzo de la codificación o el comienzo de la decodificación).
- [0087]** Como se muestra en el ejemplo de la figura 3A, para una partición de minicuñas, el bloque de profundidad 110 se divide en dos regiones, la región 112 y la región 114, mediante la línea recta 116, con el punto de inicio 118 55 ubicado en (X_s, Y_s) y el punto final 120 ubicado en (X_e, Y_e) . En el ejemplo de la figura 3A, el punto de inicio 118 puede definirse como el punto (8, 0) y el punto final 120 puede definirse como el punto (0, 8).
- 60 **[0088]** Como se muestra en el ejemplo de la figura 3B, para la partición de Contorno, un bloque de profundidad, tal como el bloque de profundidad 130, se puede dividir en dos regiones de forma irregular. En el ejemplo de la figura 3B, el bloque de profundidad 130 está dividido en la región 132 y la región 134A, 134B. Aunque los píxeles en la región 134A no están inmediatamente adyacentes a los píxeles en la región 134B, las regiones 134A y 134B están definidas para formar una única región, con el fin de predecir una PU del bloque de profundidad 130. La partición de Contorno es más flexible que la partición de minicuñas, pero puede ser relativamente más difícil de señalar. En la modalidad 65 4 de las DMM, en el caso de la norma 3D-HEVC, el patrón de partición de Contorno se obtiene implícitamente utilizando muestras de luma reconstruidas del bloque de textura cosituado.

[0089] De esta manera, un codificador de vídeo, tal como el codificador de vídeo 20 y el decodificador de vídeo 30, puede usar la línea 116, definida por el punto inicial 118 y el punto final 120, para determinar si un píxel del bloque de profundidad 110 pertenece a la región 112 (que también puede mencionarse como región "P₁") o a la región 114 (que también se puede denominar región "P₂"). Del mismo modo, un codificador de vídeo puede usar las líneas 136, 138 de la figura 5B para determinar si un píxel del bloque de profundidad 130 pertenece a la región 132 (que también se puede denominar región "P₁") o a la región 134 (que también se puede denominar región "P₂"). Las regiones "P₁" y "P₂" son convenciones de denominación predeterminada para diferentes regiones divididas de acuerdo a las DMM y, por lo tanto, la región P₁ del bloque de profundidad 110 no debería ser considerada la misma región que la región P₁ del bloque de profundidad 130.

[0090] Como se ha señalado anteriormente, cada una de las DMM puede definirse según que la DMM use partición de minicuñas o de Contorno, y según que el patrón esté explícitamente señalado o implícitamente determinado. El proceso de las DMM puede integrarse como una alternativa a las modalidades de intrapredicción especificadas en la HEVC (mostrado en la figura 2). Se puede señalar un indicador de un bit para cada PU, para especificar si se aplican las DMM o la intrapredicción convencional.

[0091] Un patrón de partición de tamaño NxN, como se muestra en las Figuras 3A y 3B, indica un bloque binario de tamaño NxN. En esta divulgación, el valor en la posición (i, j) del bloque binario de tamaño NxN se denomina valor de partición del patrón de partición en la posición (i, j), donde i, j = 0, 1, ..., N- 1. Para cada posición de un patrón de partición de tamaño NxN, el valor del dígito binario indica la partición (0 o 1) de la posición actual.

[0092] Por ejemplo, la figura 3A ilustra un ejemplo de un patrón de minicuñas en el que una línea lineal (por ejemplo, la línea recta 116) biseca el bloque de profundidad 110. Sin embargo, puede haber muchos patrones diferentes de minicuñas. Por ejemplo, en lugar de una línea lineal que comienza en (0, 8) y termina en (8, 0), como se ilustra en la figura 3A, en otro ejemplo, es posible una línea lineal que comienza en (1, 8) y termina en (8, 1). Puede haber muchos más ejemplos de patrones de minicuñas de ese tipo.

[0093] En general, el número de patrones de minicuñas puede ser una función del tamaño del bloque. Por ejemplo, los bloques de profundidad de mayor tamaño incluyen más puntos de inicio y final que los bloques de profundidad de tamaño más pequeño, lo que significa que hay más posibles patrones de minicuñas para bloques de profundidad de mayor tamaño que para bloques de profundidad de menor tamaño.

[0094] Durante la inicialización, un codificador de vídeo (por ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30) puede generar todos los patrones de minicuñas disponibles, lo que construye una lista de patrones de minicuñas. Con este fin, se generan los patrones de minicuñas para todas las combinaciones posibles de posiciones de puntos de inicio y final, y un codificador de vídeo almacena en una tabla de búsqueda las posiciones de puntos de inicio y final para cada tamaño de bloque antes del proceso de codificación. En los ejemplos en los que el codificador de vídeo 20 identifica qué patrón de minicuñas utilizar para la intrapredicción de un bloque de profundidad, el codificador de vídeo 20 puede señalar un índice de la tabla de búsqueda de los patrones de minicuñas, donde el índice identifica el patrón de minicuñas que usó el codificador de vídeo 20 para la codificación por intrapredicción del bloque de profundidad. El decodificador de vídeo 30 recibe el índice de la tabla de búsqueda de los patrones de minicuñas que el decodificador de vídeo 30 construyó durante la inicialización. El decodificador de vídeo 30 puede luego determinar el patrón de minicuñas identificado por el índice, y usar ese patrón de minicuñas para la decodificación por intrapredicción del bloque de profundidad. De esta manera, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden utilizar, respectivamente, el mismo patrón de minicuñas para la codificación y decodificación por intrapredicción.

[0095] Las posibles combinaciones de posiciones de punto inicial y final se pueden clasificar en seis categorías, según la orientación de la línea divisoria de la partición que conecta las posiciones de punto inicial y final. Por ejemplo, una línea lineal que biseca el bloque de profundidad puede extenderse desde la fila superior hasta la columna izquierda, la fila inferior o la columna derecha. Una línea lineal que biseca el bloque de profundidad puede extenderse desde la columna de la izquierda hasta la fila inferior o la columna de la derecha (la extensión a la fila superior ya está cubierta en el caso anterior). Una línea lineal que biseca el bloque de profundidad puede extenderse desde la fila inferior hasta la columna derecha (las otras ya están cubiertas en el caso anterior). En el camino, hay seis categorías para la línea lineal que biseca el bloque de profundidad. Estas seis categorías se enumeran a continuación en la Tabla 1:

Tabla 1: Orientación de la línea divisoria de la partición que conecta los puntos inicial y final

Posición del punto inicial	Posición del punto final	Orientación
Fila superior	Columna izquierda	0
Columna derecha	Fila superior	1
Fila inferior	Columna derecha	2

Posición del punto inicial	Posición del punto final	Orientación
Columna izquierda	Fila inferior	3
Fila superior	Fila inferior	4
Columna derecha	Columna izquierda	5

5 **[0096]** Para generar un patrón de partición de minicuñas de tamaño NxN, dado un punto inicial (xS, yS) y una posición de punto final (xE, yE), el codificador de vídeo 20 y el decodificador de vídeo 30 pueden generar primero un patrón temporal de partición de tamaño KxK con todas las muestras inicializadas como 0, donde K es igual a 2N para la precisión de media muestra (medio píxel) y K es igual a N para otros casos. El codificador de vídeo 20 y el decodificador de vídeo 30 pueden fijar las muestras que forman la línea divisoria de partición que conecta (xS, yS) y (xE, yE) en 1, y dividir el patrón de partición temporal en dos partes (por ejemplo, la parte A y la parte B, como se ilustra en la figura 4).

10 **[0097]** Las figuras 4A a 4F son diagramas conceptuales que ilustran ejemplos de patrones de partición temporales. Por ejemplo, las figuras 4A a 4F ilustran dos partes divididas por la línea divisoria de partición (es decir, la línea lineal que biseca el bloque de profundidad) en la generación del patrón de minicuñas.

15 **[0098]** Después de que el codificador de vídeo 20 y el decodificador de vídeo 30 dividen el patrón de partición temporal en dos partes, el codificador de vídeo 20 y el decodificador de vídeo 30 seleccionan una de las dos partes para que sea la partición 1. El codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para seleccionar cuál de las partes es la partición 1, basándose en la orientación de la línea divisoria del patrón de minicuñas. Debido a que el codificador de vídeo 20 y el decodificador de vídeo 30 están configurados para seleccionar cuál de las partes es la partición 1 de la misma manera, el codificador de vídeo 20 y el decodificador de vídeo 30 seleccionan la misma parte para que sea la partición 1. Además, debido a que la otra partición (es decir, la partición que no es la partición 1) es, por omisión, la partición 0, el codificador de vídeo 20 y el decodificador de vídeo 30 seleccionan la misma parte para que sea la partición 0.

25 **[0099]** Como se ha descrito anteriormente, y se ha indicado en la Tabla 1, puede haber seis orientaciones de la línea lineal que biseca el bloque de profundidad en dos particiones para un patrón de minicuñas. Las figuras 4A a 4F ilustran ejemplos de tales orientaciones. Debería entenderse que las figuras 4A a 4F ilustran, cada una, un ejemplo de orientaciones respectivas, y asimismo puede haber otros ejemplos. Por ejemplo, la figura 4B es para la orientación 1, y la Tabla 1 anterior indica que la orientación 1 es desde la columna derecha a la fila superior. En la figura 4B, la línea lineal comienza en (8, 5) y termina en (4, 0). En otro ejemplo de la orientación 1, la línea lineal puede comenzar en (8, 7) y terminar en (1, 0).

30 **[0100]** La tabla 2, a continuación, ilustra la manera en que el codificador de vídeo 20 y el decodificador de vídeo 30 seleccionan qué parte ha de ser la partición 1, basándose en la orientación de la frontera de la partición. Por ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden determinar primero la orientación de la frontera de la partición (es decir, la orientación de la línea lineal que biseca el bloque de profundidad). Luego, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden determinar qué partición de las dos particiones creadas por la línea lineal bisectriz debería identificarse como 1 y qué partición debería identificarse como 0.

35 **[0101]** Como ejemplo, como se indica en la Tabla 2, si la frontera de la partición de orientación es 0, entonces el codificador de vídeo 20 y el decodificador de vídeo 30 seleccionan la parte A como la partición a identificar con un 1. La figura 4A ilustra un ejemplo donde la partición etiquetada con A se identificaría como 1, y la partición etiquetada con B se identificaría como 0. Sin embargo, la figura 4A es un ejemplo de frontera de partición de orientación 1, y hay otros ejemplos de frontera de partición de orientación 1. El codificador de vídeo 20 y el decodificador de vídeo 30 pueden realizar funciones similares para determinar qué partición se identifica como 1 y qué partición se identifica como 0 para las orientaciones 1 a 5, basándose en los criterios estipulados en la Tabla 2.

Tabla 2: Selección de muestras para la partición 1

Orientación de la frontera de partición	Parte de las muestras que serán seleccionadas como 1
0	A
1	B
2	B
3	A
4	$\zeta xS + xE < K ? A: B$
5	$\zeta yS + yE < K ? A: B$

5 **[0102]** En el ejemplo ilustrado en las figuras 4A a 4F, la línea lineal comienza desde un píxel (por ejemplo, una muestra) dentro del bloque de profundidad y termina en un píxel (por ejemplo, una muestra) dentro del bloque de profundidad. En tales ejemplos, se puede considerar que la línea lineal tiene una precisión de muestra completa. Sin embargo, las técnicas descritas en esta divulgación no están limitadas de ese modo. Por ejemplo, las técnicas pueden extenderse a la precisión de media muestra o incluso a la precisión de un cuarto de muestra.

10 **[0103]** Para el caso de la precisión de media muestra, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden generar el patrón de partición de tamaño NxN, denominado *patrónB*, como la versión reducida en muestras del patrón de partición temporal de tamaño KxK ($K = 2N$) *patrónTempB*. En este ejemplo, $patrónB[i][j] = patrónTempB[m][n]$, donde $i, j = 0, 1, \dots, N-1$ y $m, n = 0, 1, \dots, 2N-1$. En esta divulgación, el patrón temporal de partición de tamaño KxK se menciona como el correspondiente patrón de minicuñas de este patrón de partición de tamaño NxN, con posición inicial (xS, yS) y posición final (xE, yE), y este patrón de minicuñas correspondiente es de tamaño $2Nx2N$, con posición inicial ($2*xS, 2*yS$) y posición final ($2*xE, 2*yE$). Debido a la precisión de media muestra, $2*xS, 2*yS, 2*xE$ y $2*yE$ son números enteros, pero xS, yS, xE y yE pueden ser números fraccionarios. La correlación entre (i, j) y (m, n) depende de la orientación de la línea divisoria de la partición. Las técnicas pueden extenderse de manera similar para una precisión de un cuarto de muestra.

20 **[0104]** A continuación se describen ejemplos de cómo generar un patrón de partición para el caso de precisión de media muestra. Con fines de ilustración, los ejemplos se describen para la orientación 0 y la orientación 1.

25 **[0105]** Las figuras 5A y 5B son diagramas conceptuales que ilustran la correlación de muestras de patrones de partición para una precisión de media muestra. Por ejemplo, las figuras 5A y 5B ilustran la correlación de muestras de patrón de partición de tamaño $2Nx2N$ con muestras de patrón de partición de tamaño NxN (por ejemplo, para una precisión de media muestra). La figura 5A ilustra el ejemplo de orientación 0, donde el punto inicial está en la fila superior y el punto final está en la columna izquierda. La figura 5B ilustra el ejemplo de la orientación 1, donde el punto inicial está en la columna derecha y el punto final está en la fila superior.

30 **[0106]** En las figuras 5A y 5B, los bloques sombreados indican las muestras del patrón de partición de muestreo reducido en frecuencia de tamaño NxN. Por ejemplo, el bloque en las figuras 5A y 5B pueden tener un tamaño de $2Nx2N$ y, en las Figuras 5A y 5B, las muestras sombreadas indican una muestra de cada dos en el bloque de tamaño $2Nx2N$, dando como resultado bloques de tamaño NxN.

35 **[0107]** En algunos ejemplos, para generar patrones de minicuñas con orientación 0, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden ciclar el punto de inicio ((xS, yS)) desde (0, 0) a $(2N-1, 0)$ y ciclar el final punto ((xE, yE)) desde (0, 0) a (0, $2N-1$) para abarcar todos los posibles puntos inicial y final para los patrones de minicuñas con orientación 0. El codificador de vídeo 20 y el decodificador de vídeo 30 también pueden correlacionar (i, j) con (m, n), donde $m = 2i$ y $n = 2j$, para la orientación 0. Para el caso de la orientación 1, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden ciclar el punto de inicio ((xS, yS)) desde $(2N-1, 0)$ hasta $(2N-1, 2N-1)$ y ciclar el punto final ((xE, yE)) desde $(2N-1, 0)$ hasta (0, 0) para abarcar todos los posibles puntos inicial y final para los patrones de minicuñas con orientación 1. El codificador de vídeo 20 y el decodificador de vídeo 30 también pueden correlacionar (i, j) con (m, n), donde $m = 2i + 1$ y $n = 2j$, para la orientación 1.

45 **[0108]** Para otros casos de orientación, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden ciclar de forma similar los puntos inicial y final en función de las orientaciones. La ecuación para la correlación para todas las orientaciones se puede generalizar como $m = 2i + desplazamientoX$ y $n = 2j + desplazamientoY$, donde $desplazamientoX$ y $desplazamientoY$ son valores de desfase, como se especifica en la Tabla 3. El codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementar la ecuación $m = 2i + desplazamientoX$ y $n = 2j + desplazamientoY$ para la correlación para cada una de las orientaciones, donde $desplazamientoX$ y $desplazamientoY$ están definidos por la Tabla 3.

Tabla 3: Determinación de desplazamientoX y desplazamientoY en función de la orientación de la línea divisoria de partición

Orientación	desplazamientoX	desplazamientoY
0	0	0
1	1	0
2	1	1
3	0	1
4	$\zeta xS + xE < K ? 0: 1$	0
5	0	$\zeta yS + yE < K ? 0: 1$

55

[0109] Como se ha descrito anteriormente, durante la inicialización, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden construir, cada uno, una lista de patrones de minicuñas que incluya puntos inicial y final para todos los patrones de minicuñas (es decir, puntos inicial y final para diferentes líneas lineales posibles que bisecan el bloque de profundidad en dos particiones para cada orientación). En algunos casos, dos patrones de minicuñas pueden ser iguales. Sin embargo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden eliminar patrones de minicuñas duplicados durante el proceso de inicialización de la lista de patrones de minicuñas, de modo que la lista de patrones de minicuñas incluya solo patrones únicos.

[0110] Al generar la lista de patrones de minicuñas, la resolución de las posiciones inicial y final, utilizadas para generar los patrones de minicuñas, depende del tamaño del bloque. Para bloques de tamaño 32x32, las posibles posiciones inicial y final están restringidas a ubicaciones con una precisión de 2 muestras. Para bloques de tamaño 16x16, se usa la precisión de la muestra completa, y para bloques de tamaños 4x4 y 8x8, se usa la precisión de media muestra. En consecuencia, el número de posibles patrones de minicuñas para diferentes tamaños de bloque puede ser diferente. En general, el número de posibles patrones de minicuñas es directamente proporcional al tamaño del bloque (es decir, para tamaños de bloque más grandes, más patrones de minicuñas, y para tamaños de bloque más pequeños, menos patrones de minicuñas).

[0111] Durante el proceso de inicialización de la lista de patrones de minicuñas, para evitar agregar patrones de minicuñas duplicados en la lista de minicuñas, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden agregar, cada uno, un patrón de minicuñas recién generado al final de la lista de minicuñas, solo cuando el patrón de minicuñas recién generado no representa el mismo patrón que cualquiera de los patrones actuales de minicuñas en la lista. Cuando se comparan dos patrones de minicuñas de tamaño NxN (es decir, PatrónA[i][j], PatrónB[i][j]), donde $i, j = 0, 1, \dots, N-1$, se considera que el PatrónA es lo mismo que el PatrónB si, para todas las combinaciones posibles de i y j en el rango de $[0, N-1]$, el PatrónA[i][j] siempre es igual al PatrónB[i][j] o el PatrónA[i][j] nunca es igual al PatrónB[i][j].

[0112] Debido a los diferentes tamaños de bloque y la diferente precisión de las posiciones de punto inicial y final, hay diferentes números y diferentes patrones de patrones de minicuñas disponibles para diferentes tamaños de bloque, como se enumera a continuación en la Tabla 4. Para tamaños de bloque más grandes, el número total de patrones de minicuñas puede ser mucho mayor, lo que puede aumentar el requisito de almacenamiento y la complejidad para la generación de patrones.

Tabla 4: Número de patrones de minicuñas disponibles en las DMM

Tamaño de bloque	Número de patrones de minicuñas disponibles
4x4	86
8x8	782
16x16	1394
32x32	1503
64x64	N/A

[0113] Se ha especificado un proceso detallado de inicialización de la lista de minicuñas en H.8.4.4.2.12, H.8.4.4.2.12.1 y H.8.4.4.2.12.2 del borrador de trabajo JCT3V-E1001. Como se ha señalado anteriormente, el documento JCT3V-E1001 se titula "3D-HEVC Draft Text 1 [Texto borrador 1 de la norma 3D-HEVC]" por Sullivan et al. y está disponible, a partir del 13 de octubre de 2014, en http://phenix.it-sudparis.eu/jct2/doc_end_user/current_document.php?id=1361.

[0114] Puede haber ciertos problemas para el diseño de patrones de minicuñas dependiente del tamaño de bloque. Por ejemplo, para tamaños grandes de bloques, el número total de patrones de minicuñas disponibles puede ser muy grande. Como se ilustra en la Tabla 4, para tamaños de bloque de 16x16, el número total de patrones de minicuñas es 1394 y, para tamaños de bloque de 32x32, el número total de patrones de minicuñas es 1503. La gran cantidad de patrones de minicuñas introduce un requisito de memoria grande, que puede no ser deseable. En otras palabras, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden necesitar una memoria de gran tamaño para almacenar todos los patrones de minicuñas. Además, para construir la lista de patrones de minicuñas, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden requerir un número indeseable de llamadas de memoria a la unidad de memoria para recuperar los patrones de minicuñas, porque hay un número relativamente grande de patrones de minicuñas, especialmente para tamaños de bloques más grandes.

[0115] Por este motivo (es decir, la gran cantidad de patrones), los patrones de minicuñas no disponen de soporte para las PU de tamaño 64x64. Por ejemplo, en el diseño de patrones de minicuñas dependiente del tamaño de bloque existente, los beneficios asociados al uso de un patrón de minicuñas para la intrapredicción de un bloque de profundidad pueden ser contrarrestados por la gran cantidad de patrones de minicuñas para bloques de profundidad de tamaño 64x64. En consecuencia, incluso aunque pueda haber beneficios asociados a la intrapredicción basada en patrones de minicuñas para bloques de tamaño 64x64, las técnicas existentes pueden no prestar soporte a dicha

intrapredicción basándose en el patrón de minicuñas para bloques de tamaño 64x64.

[0116] Como otro ejemplo de problemas con el diseño de patrones de minicuñas dependiente del tamaño de bloque existente, en la norma 3D-HEVC, el proceso de generación de un patrón de minicuñas puede ser complejo, especialmente cuando el bloque tiene un tamaño mayor. Además, la norma 3D-HEVC puede requerir que, para cada tamaño de bloque, se almacene un conjunto de patrones de minicuñas. Sin embargo, para bloques de tamaño 64x64, la intrapredicción que utiliza patrones de minicuñas puede no disponer de soporte, por lo que no es necesario almacenar ningún patrón de minicuñas para bloques de tamaño 64x64, para el diseño actual de patrones de minicuñas.

[0117] Como otro ejemplo más de problemas con el diseño de patrones de minicuñas dependiente del tamaño de bloque existente, durante la inicialización de la lista de patrones de minicuñas, se puede requerir que el codificador de vídeo 20 y el decodificador de vídeo 30 realicen un número significativo de cálculos de comparación entre dos patrones de minicuñas para evitar agregar patrones duplicados de minicuñas. Dichos cálculos de comparación aumentan la complejidad del cálculo y no son deseables para el codificador de vídeo 20 o el decodificador de vídeo 30.

[0118] Las técnicas descritas en esta descripción se refieren a la generación simplificada de patrones de partición de profundidad para la intracodificación de profundidad. Los siguientes ejemplos describen técnicas para una tal generación simplificada de patrones de partición de profundidad para la intracodificación de profundidad. Las siguientes técnicas ejemplares pueden ser realizadas por el codificador de vídeo 20 y el decodificador de vídeo 30. Por ejemplo, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden intracodificar datos de profundidad (es decir, codificar por intrapredicción o decodificar por intrapredicción, respectivamente). Además, las técnicas ejemplares se pueden realizar por separado o bien una o más técnicas se pueden realizar en combinación.

[0119] Además, por brevedad, las técnicas descritas en esta divulgación pueden describirse con respecto a un codificador de vídeo. Los ejemplos del codificador de vídeo incluyen el codificador de vídeo 20 al describir la codificación de datos de vídeo, y el decodificador de vídeo 30 al describir la decodificación de datos de vídeo. Esta divulgación también puede utilizar los términos "código" o "codificación". En este contexto, el término código puede referirse genéricamente a codificar o a decodificar y el término codificación puede referirse genéricamente a la codificación o a la decodificación. Por ejemplo, un codificador de vídeo puede codificar, lo que significa que el codificador de vídeo 20 puede codificar o el decodificador de vídeo 30 puede decodificar.

[0120] Como se ha descrito anteriormente, uno de los problemas con el diseño de patrones de minicuñas dependiente del tamaño de bloque existente es que, para bloques de mayor tamaño, el número de patrones de minicuñas se vuelve demasiado grande, lo que requiere una cantidad indeseable de memoria y aumenta la complejidad de cálculo. En las técnicas descritas en esta divulgación, en lugar de confiar en los patrones de minicuñas para bloques de profundidad de mayor tamaño, el patrón de minicuñas para el bloque de profundidad de mayor tamaño puede determinarse a partir de un patrón de minicuñas para bloques de menor tamaño. De esta manera, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden almacenar los patrones de minicuñas (por ejemplo, patrones de partición) para bloques de menor tamaño y almacenar menos patrones de minicuñas, o ninguno, para bloques de mayor tamaño, porque el codificador de vídeo 20 y el decodificador de vídeo 30 pueden determinar el patrón de minicuñas para el bloque de profundidad de mayor tamaño a partir de un patrón de minicuñas de los patrones de minicuñas para bloques de menor tamaño.

[0121] Como ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para construir una lista de patrones de partición (por ejemplo, la lista de patrones de minicuñas) para bloques de un primer tamaño de bloque. La lista de patrones de partición puede incluir uno o más patrones de partición asociados a bloques del primer tamaño de bloque. Para la codificación por intrapredicción de un bloque de profundidad de un segundo tamaño de bloque, mayor que el primer tamaño de bloque, el codificador de vídeo 20 puede codificar por intrapredicción el bloque de profundidad del segundo tamaño, usando un patrón de partición (por ejemplo, un patrón de minicuñas) determinado a partir de los patrones de partición para bloques del primer tamaño de bloque. En otras palabras, el codificador de vídeo 20 puede codificar por intrapredicción el bloque de profundidad del segundo tamaño utilizando un patrón de partición asociado a bloques del primer tamaño. Para la modalidad de modelado de profundidad (DMM) 1, el codificador de vídeo 20 puede señalar un índice de la lista de patrones de partición que incluye los uno o más bloques asociados de patrones de partición del primer tamaño de bloque.

[0122] Para la decodificación por intrapredicción del bloque de profundidad del segundo tamaño de bloque, el decodificador de vídeo 30 puede recibir el índice de la lista de patrones de partición que incluye uno o más patrones de partición asociados a bloques del primer tamaño de bloque, y puede determinar un patrón de partición asociado a bloques del primer tamaño a partir del índice. Nuevamente, el patrón de partición está asociado a bloques de un primer tamaño de bloque que es más pequeño que el segundo tamaño de bloque del bloque de profundidad que se está decodificando. El decodificador de vídeo 30 puede entonces determinar un patrón de partición para el bloque de profundidad del segundo tamaño de bloque a partir del patrón de partición determinado. El decodificador de vídeo 30 puede decodificar por intrapredicción el bloque de profundidad del segundo tamaño basándose en el patrón de partición determinado.

[0123] Como se ha descrito anteriormente, el almacenamiento de patrones de partición (por ejemplo, patrones de

minicuñas) para bloques de mayor tamaño puede ser indeseable debido a la cantidad relativamente grande de patrones de partición que están disponibles para bloques de mayor tamaño. Al determinar el patrón de partición para bloques de mayor tamaño en función de un patrón de partición a partir de patrones de partición para bloques de tamaño más pequeño, se puede reducir la cantidad de patrones de partición que se deben almacenar para bloques de mayor tamaño. Por ejemplo, en el ejemplo anterior, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden construir una lista de patrones de partición que incluya uno o más patrones de partición asociados a bloques del primer tamaño de bloque, y usar uno de los patrones de partición de la lista de patrones de partición para determinar el patrón de partición para un bloque de profundidad de un segundo tamaño más grande. En consecuencia, si el codificador de vídeo 20 y el decodificador de vídeo 30 construyeran la lista de patrones de partición para los patrones de partición del segundo tamaño, el número de patrones de partición en la lista de patrones de partición se reduciría debido a que algunos de los patrones de partición pueden determinarse a partir de patrones de partición para tamaños de bloque más pequeños.

[0124] En algunos ejemplos, el codificador de vídeo 20 y el decodificador de vídeo 30 ni siquiera pueden construir patrones de lista de particiones para ciertos tamaños de bloques. En tales ejemplos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden depender de patrones de partición asociados a bloques de tamaño más pequeño para determinar los patrones de partición para los bloques de profundidad de los tamaños de bloque para los cuales el codificador de vídeo 20 y el decodificador de vídeo 30 construyeron un patrón de lista de particiones.

[0125] Además, debido a que puede haber menos patrones de partición, o ninguno, en una lista de patrones de partición para tamaños de bloque más grandes, puede haber una reducción en la cantidad de memoria necesaria para almacenar los patrones de partición. Además, debido a que puede haber menos patrones de partición, o ninguno, en una lista de patrones de partición para tamaños de bloque más grandes, la complejidad de cálculo de la generación de los patrones de partición y la complejidad de la comparación de patrones de partición para asegurar que no haya duplicados también pueden reducirse.

[0126] Como se ha descrito anteriormente, al determinar un patrón de partición para un bloque de profundidad de un tamaño más grande basándose en un patrón de partición para un bloque de tamaño más pequeño, las técnicas descritas en esta divulgación pueden reducir los requisitos de memoria y la complejidad. En algunos ejemplos, el bloque de tamaño más pequeño puede ser un bloque dentro del bloque de profundidad de tamaño más grande. Sin embargo, las técnicas descritas en esta divulgación no están limitadas de ese modo. En algunos ejemplos, el bloque de menor tamaño no tiene por qué ser necesariamente un bloque real que esté siendo codificado o decodificado por intrapredicción. Más bien, el bloque de menor tamaño puede ser un bloque conceptual cuyos patrones de partición se utilizan para determinar un patrón de partición para un bloque de mayor tamaño.

[0127] Como ejemplo, supongamos que el bloque de profundidad que está siendo codificado o decodificado por intrapredicción es un bloque de profundidad de tamaño 32x32. En este ejemplo, el codificador de vídeo 20 puede señalar un índice de una lista de patrones de partición para bloques de tamaño 16x16. El decodificador de vídeo 30 puede recibir el índice de la lista de patrones de partición para bloques de tamaño 16x16 y determinar el patrón de partición entre los patrones de partición para bloques de tamaño 16x16. En este ejemplo, tanto el codificador de vídeo 20 como el decodificador de vídeo 30 pueden configurarse para determinar el patrón de partición para el bloque de profundidad de tamaño 32x32 a partir del patrón de partición determinado para bloques de tamaño 16x16. Como ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden aumentar la frecuencia de muestreo del patrón de partición determinado para bloques de tamaño 16x16, para determinar el patrón de partición para el bloque de profundidad de tamaño 32x32.

[0128] Aunque el bloque de menor tamaño, cuya lista de patrones de partición usan el codificador de vídeo 20 y el decodificador de vídeo 30 para determinar el patrón de partición para el bloque de mayor tamaño, no necesita ser parte del bloque de mayor tamaño, las técnicas descritas en esta divulgación se describen con ejemplos donde el bloque de menor tamaño es un bloque dentro del bloque de mayor tamaño. Por ejemplo, en un ejemplo para la codificación de profundidad (por ejemplo, la codificación por intrapredicción de un bloque de profundidad), los patrones de partición son independientes de los tamaños de bloque.

[0129] Debería entenderse que, en las técnicas descritas en esta divulgación, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden determinar un patrón de partición para un bloque de tamaño más pequeño como una etapa intermedia para determinar un patrón de partición para un bloque de tamaño más grande. Por ejemplo, el bloque de tamaño más pequeño puede no estar codificado o decodificado por intrapredicción, en función del patrón de partición del bloque de tamaño más pequeño. Más bien, el bloque de tamaño más pequeño se puede codificar o decodificar por intrapredicción como parte de la intrapredicción del bloque de tamaño más grande.

[0130] Como ejemplo, supóngase que el bloque de profundidad actual (por ejemplo, la PU de profundidad actual) tiene un tamaño de NxN. En este ejemplo, el bloque de profundidad actual incluye un número entero de bloques de tamaño MxM (donde $M < N$). Como ejemplo, si el bloque de profundidad actual (por ejemplo, el bloque de profundidad a codificar o decodificar por intrapredicción) es un bloque de tamaño 32x32, hay cuatro bloques de tamaño 16x16 dentro del bloque de tamaño 32x32, o 64 bloques de tamaño 4x4 dentro del bloque de tamaño 32x32.

[0131] En algunos ejemplos, el codificador de vídeo 20 puede señalar información que identifica un bloque específico de tamaño $M \times M$ (por ejemplo, donde M es igual a 4) y un patrón de partición basado en líneas (por ejemplo, un patrón de minicuñas que define una línea lineal que biseca el bloque de tamaño $M \times M$ en dos particiones). El decodificador de vídeo 30 puede recibir la información que identifica el bloque de tamaño $M \times M$ y el patrón de partición basado en líneas. Basándose en el patrón de partición basado en líneas del bloque específico de tamaño $M \times M$, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden determinar (por ejemplo, obtener) la partición basada en líneas para toda la PU de tamaño $N \times N$. En otras palabras, según el patrón de partición para un bloque de tamaño $M \times M$, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden determinar un patrón de partición para todo el bloque de profundidad de tamaño $N \times N$, donde el bloque de profundidad de tamaño $N \times N$ incluye el bloque de tamaño $M \times M$.

[0132] En un ejemplo, el patrón de partición basado en líneas puede ser un patrón de minicuñas de la DMM 1 aplicado al bloque específico de tamaño $M \times M$ como en la norma 3D-HEVC actual. Por ejemplo, el codificador de vídeo 20 puede señalar información que indica que la modalidad de modelado de profundidad (DMM) es 1, señalar información que identifica un bloque de tamaño $M \times M$ dentro del bloque de tamaño $N \times N$ y señalar información usada para identificar el patrón de partición para el bloque de tamaño $M \times M$. (por ejemplo, un valor de índice de una lista de patrones de partición para bloques de tamaño $M \times M$). El decodificador de vídeo 30 puede recibir información que indica que la DMM es 1, determinar el bloque de tamaño $M \times M$ dentro del bloque de tamaño $N \times N$ basándose en la información recibida que identifica el bloque de tamaño $M \times M$ y determinar el patrón de partición para el bloque de tamaño $M \times M$ (por ejemplo, basándose en el valor de índice de la lista de patrones de partición para bloques de tamaño $M \times M$). El decodificador de vídeo 30 puede luego determinar el patrón de partición para todo el bloque de tamaño $N \times N$ basándose en el patrón de partición determinado para el bloque de tamaño $M \times M$.

[0133] Como se describe en el presente documento, un bloque "entero" puede incluir el bloque completo, incluyendo todo subbloque que pueda estar incluido en el bloque. En consecuencia, para un bloque que tiene cuatro subbloques, el bloque entero puede referirse a todo el bloque, incluidos los cuatro subbloques.

[0134] Por ejemplo, el codificador de vídeo 20 puede determinar un patrón de partición basado en líneas para un subbloque (por ejemplo, un patrón de minicuñas que define una línea lineal que biseca un bloque de tamaño $M \times M$ en dos particiones) de un bloque (por ejemplo, un bloque de tamaño $N \times N$ que es una PU de profundidad) para la intracodificación de profundidad (codificación por intrapredicción), donde el subbloque es más pequeño que el bloque. En algunos ejemplos, el codificador de vídeo 20 puede determinar el patrón de partición basado en líneas para el subbloque del bloque sin basar el patrón determinado de partición basado en líneas en un tamaño del bloque (es decir, independiente del tamaño del bloque). El codificador de vídeo 20 puede determinar un patrón de partición basado en líneas para el bloque basándose en el patrón de partición basado en líneas para el subbloque. El codificador de vídeo 20 puede intracodificar (codificar por intrapredicción) el bloque en función del patrón de partición basado en líneas, determinado para el bloque. El codificador de vídeo 20 puede señalar información que indica el patrón de partición basado en líneas para el subbloque (por ejemplo, codificar información que indica el patrón de partición basado en líneas para el subbloque). En algunos ejemplos, el codificador de vídeo 20 puede señalar información utilizada para identificar el subbloque dentro del bloque cuyo patrón de partición basado en líneas determinó el codificador de vídeo 20; sin embargo, esto no se requiere en todos los ejemplos.

[0135] El codificador de vídeo 20 puede no señalar información que indique el patrón de partición basado en líneas para el bloque. Por ejemplo, el codificador de vídeo 20 puede evitar la codificación de, o no codificar, información que indica el patrón de partición basado en líneas para el bloque. Más bien, el codificador de vídeo 20 puede señalar información que indica el patrón de partición basado en líneas para el subbloque.

[0136] El decodificador de vídeo 30 puede recibir información que indica un patrón de partición basado en líneas para un subbloque (por ejemplo, decodificar información que indica un patrón de partición basado en líneas para un subbloque) de un bloque para la intradecodificación de profundidad (por ejemplo, para la decodificación por intrapredicción). El decodificador de vídeo 30 puede determinar un patrón de partición basado en líneas para el bloque basándose en el patrón de partición basado en líneas para el subbloque. El decodificador de vídeo 30 puede intradecodificar (decodificar por intrapredicción) el bloque basándose en el patrón de partición basado en líneas, determinado para el bloque. En algunos ejemplos, el decodificador de vídeo 30 puede recibir información utilizada para identificar el subbloque dentro del bloque cuyo patrón de partición basado en líneas fue recibido por el decodificador de vídeo 30; sin embargo, esto no se requiere en todos los ejemplos. El decodificador de vídeo 30 puede determinar el patrón de partición basado en líneas para el bloque sin recibir información que indique el patrón de partición basado en líneas para el bloque. En cambio, el decodificador de vídeo 30 puede determinar el patrón de partición basado en líneas a partir del subbloque. Además, el decodificador de vídeo 30 puede determinar el patrón de partición basado en líneas para el bloque sin basar el patrón determinado de partición basado en líneas en un tamaño del bloque (es decir, independiente del tamaño del bloque).

[0137] Debería entenderse que el subbloque de tamaño $M \times M$ no tiene por qué ser necesariamente un bloque que encaje dentro de un bloque de tamaño $N \times N$ más grande. Más bien, el subbloque de tamaño $M \times M$ puede ser un bloque conceptual cuyos patrones de partición basados en líneas se usan con fines de determinar un patrón de partición basado en líneas para el bloque de tamaño $N \times N$. En otras palabras, en el ejemplo anterior, el codificador de vídeo 20

puede determinar un patrón de partición (por ejemplo, un patrón de partición basado en líneas como los patrones de minicuñas) para un bloque de profundidad de un primer tamaño (por ejemplo, 32x32) basándose en un patrón de partición para un bloque de un segundo tamaño, más pequeño (por ejemplo, 16x16). El codificador de vídeo 20 puede codificar por intrapredicción el bloque de profundidad basándose en el patrón de partición determinado. El decodificador de vídeo 30 puede determinar de manera similar un patrón de partición para un bloque de profundidad de un primer tamaño (por ejemplo, 32x32) basándose en un patrón de partición para bloques de un segundo tamaño, más pequeño (por ejemplo, 16x16). Nuevamente, este bloque de menor tamaño no necesita ser un bloque real en la imagen o dentro del bloque de profundidad, sino que, en cambio, es un bloque conceptual cuyos patrones de partición se usan para determinar un patrón de partición para un bloque más grande.

[0138] En algunos ejemplos, el codificador de vídeo 20 señala información que identifica el patrón de partición para bloques del segundo tamaño, y el decodificador de vídeo 30 determina el patrón de partición para bloques del segundo tamaño basándose en la información señalizada. Por ejemplo, el codificador de vídeo 20 puede señalar un índice para una lista de patrones de partición, para patrones de partición del segundo tamaño, donde el patrón de partición identificado por el índice es el patrón de partición que el codificador de vídeo 20 utilizó para determinar el patrón de partición para el bloque de profundidad del primer tamaño. El decodificador de vídeo 30 puede recibir el índice de la lista de patrones de partición, para patrones de partición del segundo tamaño, y determinar el patrón de partición para el bloque de profundidad del primer tamaño basándose en el patrón de partición para bloques del segundo tamaño.

[0139] En los ejemplos anteriores, el bloque de menor tamaño es un bloque conceptual y no necesariamente un bloque de la imagen que incluye el bloque de profundidad, o un bloque dentro del bloque de profundidad. Sin embargo, en algunos ejemplos, el bloque más pequeño puede ser un bloque dentro del bloque de mayor tamaño. En ejemplos donde el bloque de menor tamaño es un bloque dentro del bloque de mayor tamaño (por ejemplo, el bloque de tamaño 16x16 es uno de los cuatro bloques de tamaño 16x16 dentro del bloque de tamaño 32x32), el codificador de vídeo 20 puede señalar información que identifica la ubicación del bloque de menor tamaño en el bloque de mayor tamaño. El decodificador de vídeo 30 puede entonces determinar el patrón de partición para el bloque de mayor tamaño basándose en el patrón de partición para el bloque de menor tamaño y en la posición del bloque de menor tamaño dentro del bloque de mayor tamaño. Por ejemplo, el decodificador de vídeo 30 puede extender la línea lineal que biseca el bloque de tamaño más pequeño hacia afuera, hasta que la línea lineal encuentre los bordes del bloque de mayor tamaño. La bisección resultante del bloque de mayor tamaño puede ser el patrón de partición para el bloque de mayor tamaño, determinado a partir del patrón de partición del bloque de menor tamaño.

[0140] En otras palabras, el decodificador de vídeo 30 puede extender una línea lineal del patrón de partición para el subbloque hasta las fronteras del bloque de profundidad. La línea lineal resultante puede ser el patrón de partición para el bloque de profundidad utilizado para decodificar por intrapredicción el bloque de profundidad. El codificador de vídeo 20 puede extender de manera similar una línea lineal del patrón de partición para el subbloque hasta las fronteras del bloque de profundidad, y la línea lineal resultante puede ser la partición para el bloque de profundidad, usada para codificar por intrapredicción el bloque de profundidad.

[0141] En algunos ejemplos, el codificador de vídeo 20 puede no necesitar señalar información que identifique la posición del bloque de tamaño más pequeño. En tales ejemplos, el decodificador de vídeo 30 puede preconfigurarse para usar un bloque de menor tamaño establecido como el bloque desde el cual el decodificador de vídeo 30 extiende el patrón de partición para determinar el patrón de partición para el bloque de profundidad de mayor tamaño.

[0142] Por ejemplo, el codificador de vídeo 20 puede señalar el índice del bloque específico de tamaño $M \times M$ (es decir, el subbloque), que recibe el decodificador de vídeo 30, con un índice relativo horizontal y vertical (i, j) , con i y j en el rango de 0 a $N/M-1$, inclusive. En este ejemplo, la posición superior izquierda del bloque de tamaño $M \times M$ es $(M*i, N*j)$. De esta manera, el decodificador de vídeo 30 puede determinar la posición del bloque de tamaño $M \times M$ y además determinar el patrón de partición para todo el bloque de tamaño $N \times N$ al extender la línea lineal, como se describe con más detalle a continuación.

[0143] El codificador de vídeo 20 y el decodificador de vídeo 30 pueden codificar o decodificar, respectivamente, la posición (i, j) del bloque de tamaño $M \times M$, bien con la modalidad de omisión o bien con el modelado de contexto. En otras palabras, el codificador de vídeo 20 puede codificar la posición (i, j) utilizando la modalidad de omisión o el modelado de contexto para identificar el subbloque cuyo patrón de partición basado en líneas fue determinado, y el decodificador de vídeo 30 puede decodificar la posición (i, j) usando la modalidad de omisión o el modelado de contexto para identificar el subbloque cuyo patrón de partición basado en líneas fue recibido.

[0144] Como alternativa, o adicionalmente, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden preconfigurarse para fijar el bloque específico de tamaño $M \times M$ para que comience siempre en el bloque de tamaño $M \times M$ de la frontera izquierda o en el bloque de tamaño $M \times M$ de la frontera inferior de la PU actual. En este caso, el codificador de vídeo 20 puede señalar un indicador y solo un desplazamiento es señalado adicionalmente por el codificador. Por ejemplo, en este ejemplo, el codificador de vídeo 20 puede señalar un indicador y un desplazamiento para identificar el subbloque cuyo patrón de partición basado en líneas se determinó, y el decodificador de vídeo 30 puede recibir un indicador y un desplazamiento para identificar el subbloque cuyo patrón de partición basado en líneas se recibió.

[0145] En algunos casos, en lugar de señalar el índice horizontal y vertical en la unidad de tamaño MxM dentro de un bloque de tamaño NxN, el codificador de vídeo 20 puede identificar el bloque de tamaño MxM por una estructura de árbol cuádruple, en donde cada representación de nivel tiene cero o más "0" y un "1" y termina y avanza a un nivel inferior una vez que se alcanza un "1". En este ejemplo, el codificador de vídeo 20 puede no necesitar señalar información para identificar el subbloque y el decodificador de vídeo 30 puede no necesitar recibir información para identificar el subbloque. Por ejemplo, el decodificador de vídeo 30 puede usar de manera similar la estructura de árbol cuádruple para determinar el bloque de tamaño MxM para el cual el decodificador de vídeo 30 determinó un patrón de partición, para determinar el patrón de partición para el bloque de profundidad de tamaño NxN.

[0146] En algunos ejemplos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden extender el patrón de partición basado en líneas del bloque específico de tamaño MxM hasta un bloque de tamaño NxN más grande, de manera que los patrones de partición de todos los bloques de tamaño MxM dentro del bloque de tamaño NxN, si están disponibles, formen juntos una partición basada en líneas para toda la PU (es decir, todo el bloque de profundidad). Por ejemplo, en la esquina superior derecha de un bloque de tamaño MxM en particular está la esquina inferior izquierda de otro bloque de tamaño MxM dentro de un bloque de tamaño NxN. En algunos ejemplos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden extender la línea lineal (por ejemplo, la línea de partición) que biseca un bloque de tamaño MxM particular a través del otro bloque conectado a la esquina superior derecha del bloque de tamaño MxM particular. De manera similar, en la esquina inferior izquierda del bloque de tamaño MxM particular se encuentra la esquina superior derecha de otro bloque más de tamaño MxM dentro del bloque de tamaño NxN. En algunos ejemplos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden extender la línea lineal (por ejemplo, la línea de partición) que biseca un bloque de tamaño MxM particular a través del otro bloque conectado a la esquina inferior izquierda del bloque de tamaño MxM particular. De esta manera, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden extender la partición basada en líneas de un bloque específico de tamaño MxM a toda la PU.

[0147] Como se ha descrito anteriormente, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para extender la línea lineal que biseca el bloque de tamaño más pequeño, de manera que la línea lineal biseque el bloque de tamaño más grande para determinar el patrón de partición para el bloque de profundidad de tamaño más grande. En algunos ejemplos, para extender el patrón de partición específico de tamaño MxM basado en líneas a la PU actual (es decir, a todo el bloque de profundidad de mayor tamaño), el codificador de vídeo 20 y el decodificador de vídeo 30 pueden obtener primero la función de línea divisoria de partición $y=a*x+b$, por (M^*i, M^*j) y las posiciones de punto inicial y final del patrón de partición de tamaño MxM, donde a y b representan, respectivamente, la pendiente y la intersección de la línea divisoria de partición. En otras palabras, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden determinar la ecuación lineal de la línea lineal basándose en la fórmula de pendiente-línea. Con la función de línea divisoria de partición, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden obtener un patrón de partición de tamaño NxN (denominado *patrónB*), que es un bloque binario de tamaño NxN, como $patrónB[x][y] = (y - a*x) < b ? 1 : 0$, donde $x, y = 0,1,... N-1$. En un ejemplo, además, a y b se redondean a números enteros.

[0148] Cuando la modalidad 3 o 4 de las DMM está habilitada, el codificador de vídeo 20 puede señalar un bloque específico de tamaño MxM y el decodificador de vídeo 30 puede recibirlo. Además, dentro del bloque de tamaño MxM, el codificador de vídeo 20 señala el patrón de partición basado en líneas (por ejemplo, el patrón de minicuñas) de una manera similar a como el codificador de vídeo 20 señala el patrón de minicuñas en la modalidad 3 de las DMM para una PU de tamaño MxM (es decir, un bloque de tamaño MxM), como en la norma 3D-HEVC actual. En este caso, el codificador de vídeo 20 puede señalar un índice de subconjunto de minicuñas en relación con una PU de tamaño MxM (es decir, un bloque de tamaño MxM).

[0149] Como se ha descrito anteriormente, con las técnicas descritas en esta divulgación, puede haber una reducción en el número de patrones de partición que se necesitan para bloques de mayor tamaño. Por ejemplo, en las técnicas descritas en esta divulgación, solamente hasta N/M (o 2xN/M) bloques de tamaño MxM pueden necesitar una partición (es decir, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden necesitar construir listas de particiones que incluyan solo hasta N/M (o 2xN/M) bloques de tamaño MxM). En consecuencia, el codificador de vídeo 20 puede necesitar solo señalar patrones de partición para hasta N/M (o 2xN/M) bloques de tamaño MxM, y obtener o extender hasta N/M (o 2xN/M) bloques de tamaño MxM. De manera similar, el decodificador de vídeo 30 puede necesitar solo recibir patrones de partición para hasta N/M (o 2xN/M) bloques de tamaño MxM, y obtener o extender hasta N/M (o 2xN/M) bloques de tamaño MxM.

[0150] Además, como se ha descrito anteriormente, las técnicas descritas en esta divulgación pueden reducir el número de patrones de partición que necesitan ser almacenados y reducir la complejidad de la implementación de la intrapredicción utilizando patrones de partición. En consecuencia, las técnicas pueden superar los problemas prestando soporte a la partición basada en líneas (por ejemplo, la partición de minicuñas) para bloques de profundidad de tamaño 64x64. En otras palabras, en lugar de prestar soporte solo a una partición basada en líneas de tamaño hasta 32x32, como en la norma 3D-HEVC actual, utilizando las técnicas descritas en esta divulgación, el patrón de minicuñas se extiende a una PU de tamaño NxN, que puede ser de tamaño 64x64.

[0151] Además de describir técnicas para usar patrones de partición basados en líneas para bloques de profundidad de tamaño $N \times N$ (es decir, no limitados a bloques de profundidad de tamaño menor o igual a 32×32), la divulgación describe técnicas para aumentar la eficacia de la implementación de la codificación y decodificación por intrapredicción para particiones basadas en líneas (por ejemplo, particiones de minicuñas). Por ejemplo, en el codificador de vídeo 20, para acelerar el proceso de búsqueda de minicuñas para una PU de profundidad de tamaño $N \times N$ (es decir, un bloque de profundidad) en la DMM 1, cuando las muestras superior izquierda, superior derecha, inferior izquierda e inferior derecha de la PU de profundidad actual (es decir, el bloque de profundidad) en la imagen de profundidad original tienen el mismo valor, el codificador de vídeo 20 puede omitir el proceso de búsqueda del patrón de minicuñas y el codificador de vídeo 20 puede configurarse para no seleccionar la DMM 1 para la PU actual (es decir, el bloque de profundidad actual).

[0152] Como otro ejemplo, alternativamente o adicionalmente, el codificador de vídeo 20 puede verificar los valores de múltiples PU de profundidad actuales y adyacentes. El codificador de vídeo 20 puede determinar si el proceso de búsqueda se omite y la DMM 1 se basa en los valores de las PU de profundidad actuales y adyacentes, y puede seleccionar la DMM1 para la PU actual según la determinación. De forma alternativa o adicional, el codificador de vídeo 20 puede omitir la búsqueda del patrón de minicuñas y seleccionar un patrón de partición de una muestra (por ejemplo, la muestra superior izquierda pertenece a una partición diferente a todas las demás muestras) cuando las muestras superior izquierda, superior derecha, inferior izquierda e inferior derecha de la PU de profundidad actual en la imagen de profundidad original tienen el mismo valor.

[0153] En algunos ejemplos, en la modalidad DMM 3, si el subconjunto de minicuñas de tamaño $N \times N$, especificado por la intramodalidad de luma de textura de ubicación conjunta, está vacío, el codificador de vídeo 20 puede omitir la modalidad DMM 3 para la PU actual. Que la intramodalidad de luma de textura cosituada esté vacía significa que el componente de luma del bloque de textura cosituada no está codificado o decodificado por intrapredicción. Por ejemplo, si el componente de luma del bloque de textura cosituada está codificado por intrapredicción, no hay ninguna intramodalidad de luma disponible. De forma alternativa o adicional, el decodificador de vídeo 30 puede estar restringido de manera que si el subconjunto de minicuñas de tamaño $N \times N$ especificado por la intramodalidad de luma de textura cosituada está vacío, el índice de modalidad de las DMM decodificado por el decodificador de vídeo 30 nunca puede ser la modalidad 3 de las DMM para la PU actual.

[0154] Como se ha descrito anteriormente, para la DMM 3, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden determinar el patrón de partición para el bloque de profundidad basándose en el bloque de textura cosituado. Sin embargo, las técnicas descritas en esta divulgación no están limitadas de ese modo. En algunos ejemplos, en la modalidad DMM 3, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden no obtener el patrón de partición de minicuñas por el bloque de luminancia que está cosituado con la PU actual de tamaño $N \times N$. En cambio, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden determinar el patrón de partición basado en líneas por el bloque de luma que está cosituado con el bloque actual de tamaño $M \times M$.

[0155] Para un patrón de minicuñas que utiliza la precisión de media muestra de las posiciones de punto inicial / final de una PU de tamaño $N \times N$, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden determinar a qué partición pertenece la posición (m, n) en función de la partición a la que pertenece la posición de decisión $(2m, 2n)$ en una PU de tamaño $2N \times 2N$ con el correspondiente patrón de minicuñas. El codificador de vídeo 20 y el decodificador de vídeo 30 pueden determinar a qué partición pertenece la posición (m, n) sin desplazar la posición de decisión $(2m, 2n)$ en el bloque de tamaño $2N \times 2N$, con un vector de desplazamiento de $(\text{desplazamientoX}, \text{desplazamientoY})$, en donde desplazamientoX o desplazamientoY es igual a 0 o 1 en función de varias condiciones, como en la norma 3D-HEVC actual.

[0156] La figura 6 es un diagrama de bloques que ilustra un ejemplo del codificador de vídeo 20 que puede implementar técnicas para la codificación de profundidad. El codificador de vídeo 20 puede realizar la intracodificación y la intercodificación de bloques de vídeo dentro de fragmentos de vídeo. La intracodificación (es decir, la codificación por intrapredicción) se basa en la predicción espacial para reducir o eliminar la redundancia espacial en el vídeo dentro de una trama o imagen de vídeo dada. La intercodificación (es decir, la codificación por interpredicción) se basa en la predicción temporal para reducir o eliminar la redundancia temporal en el vídeo dentro de tramas o imágenes adyacentes de una secuencia de vídeo. La intramodalidad (modalidad I) puede referirse a cualquiera de varias modalidades de codificación de base espacial. Las intermodalidades, tales como la predicción unidireccional (modalidad P) o la bipredicción (modalidad B), pueden referirse a cualquiera de varias modalidades de codificación de base temporal.

[0157] Como se ha indicado anteriormente, el codificador de vídeo 20 puede ser adaptado para realizar una codificación de vídeo de múltiples visualizaciones. En algunos casos, el codificador de vídeo 20 puede configurarse para codificar la norma HEVC de múltiples vistas, de modo que cada vista en una instancia temporal pueda ser procesada por un decodificador, tal como el decodificador de vídeo 30. Para la norma HEVC-3D, además de codificar mapas de textura (es decir, valores de luma y croma) para cada vista, el codificador de vídeo 20 puede codificar adicionalmente un mapa de profundidad para cada vista.

[0158] En cualquier caso, como se muestra en la figura 6, el codificador de vídeo 20 recibe un bloque de vídeo actual

dentro de una trama de vídeo a codificar. En el ejemplo de la figura 6, el codificador de vídeo 20 incluye los datos de memoria de vídeo 39, la unidad de selección de modalidad 40, la memoria de imágenes de referencia 64 (también mencionada como almacén temporal de imágenes decodificadas (DPB) 64), el sumador 50, la unidad de procesamiento de transformación 52, la unidad de cuantización 54 y la unidad de codificación por entropía 56. A su vez, la unidad de selección de modalidad 40 incluye la unidad de compensación de movimiento 44, la unidad de estimación de movimiento 42, la unidad de intrapredicción 46 y la unidad de partición 48. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 incluye también la unidad de cuantización inversa 58, la unidad de transformación inversa 60 y el sumador 62. También se puede incluir un filtro de eliminación de bloques (no mostrado en la figura 6) para filtrar fronteras de bloques, para eliminar distorsiones de efecto pixelado del vídeo reconstruido. Si se desea, el filtro de eliminación de bloques filtrará habitualmente la salida del sumador 62. También pueden usarse filtros adicionales (en el bucle o tras el bucle), además del filtro de eliminación de bloques. Dichos filtros no se muestran por razones de brevedad pero, si se desea, pueden filtrar la salida del sumador 50 (como un filtro en el bucle).

[0159] Como se muestra en la figura 6, la memoria de datos de vídeo 39 recibe datos de vídeo que se usan para codificar un bloque de vídeo actual dentro de una imagen de vídeo. La memoria de datos de vídeo 39 puede almacenar datos de vídeo para ser codificados por los componentes del codificador de vídeo 20 (por ejemplo, configurada para almacenar datos de vídeo) o almacenar datos de vídeo que se usarán para codificar imágenes de vídeo. Por ejemplo, la memoria de datos de vídeo 39 puede almacenar patrones de partición (por ejemplo, patrones de partición de minicuñas) para bloques de diferentes tamaños. La memoria de datos de vídeo 39 puede almacenar los patrones de partición como parte del proceso de inicialización implementado por el codificador de vídeo 20.

[0160] En algunos ejemplos, los datos de vídeo almacenados en la memoria de datos de vídeo 39 pueden obtenerse, por ejemplo, a partir del origen de vídeo 18. La memoria de imágenes de referencia 64 (también mencionada como memoria intermedia de imágenes decodificadas (DPB)) almacena datos de vídeo de referencia para su uso en la codificación de datos de vídeo por parte del codificador de vídeo 20 (por ejemplo, en modalidades de intracodificación o modalidades de intercodificación). La memoria de datos de vídeo 39 y la memoria de imágenes de referencia 64 pueden ser formadas por cualquiera entre varios dispositivos de memoria, tales como memoria dinámica de acceso aleatorio (DRAM), incluyendo DRAM síncrona (SDRAM), RAM magnetorresistiva (MRAM), RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. La memoria de datos de vídeo 39 y la memoria de imágenes de referencia 64 pueden ser proporcionadas por el mismo dispositivo de memoria o por dispositivos de memoria independientes. En diversos ejemplos, la memoria de datos de vídeo 39 puede estar en un chip con otros componentes del codificador de vídeo 20, o fuera de chip con respecto a esos componentes.

[0161] Durante el proceso de codificación, el codificador de vídeo 20 recibe una trama o un fragmento de vídeo a codificar. La trama o el fragmento pueden dividirse en múltiples bloques de vídeo. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 llevan a cabo la codificación interpredictiva del bloque de vídeo recibido con respecto a uno o más bloques en una o más tramas de referencia para proporcionar predicción temporal. Como alternativa, la unidad de intrapredicción 46 puede llevar a cabo una codificación intrapredictiva (por ejemplo, codificación por intrapredicción) del bloque de vídeo recibido con respecto a uno o más bloques vecinos en la misma trama o el mismo fragmento que el bloque a codificar, para proporcionar predicción espacial. El codificador de vídeo 20 puede llevar a cabo múltiples pases de codificación, por ejemplo, para seleccionar una modalidad adecuada de codificación para cada bloque de datos de vídeo.

[0162] Además, la unidad de partición 48 puede dividir bloques de datos de vídeo en subbloques, basándose en la evaluación de los anteriores esquemas de partición en los anteriores pases de codificación. Por ejemplo, la unidad de partición 48 puede dividir inicialmente una trama o un fragmento en varias LCU, y dividir cada una de las LCU en varias sub-CU, basándose en un análisis de velocidad-distorsión (por ejemplo, una optimización de velocidad-distorsión). La unidad de selección de modalidad 40 puede producir además una estructura de datos de árbol cuádruple, indicativa de la división de una LCU en las sub-CU. Las CU de nodos hojas del árbol cuádruple pueden incluir una o más PU y una o más TU.

[0163] La unidad de selección de modalidad 40 puede seleccionar una de las modalidades de codificación (intracodificación o intercodificación), por ejemplo, en función de los resultados erróneos, y proporcionar el bloque intracodificado o intercodificado resultante al sumador 50 para generar datos de bloques residuales, y al sumador 62 para reconstruir el bloque codificado, para su uso como una trama de referencia. La unidad de selección de modalidad 40 también puede proporcionar elementos sintácticos, tales como vectores de movimiento, indicadores de intramodalidad, información de partición y otra información sintáctica de este tipo, a la unidad de codificación por entropía 56.

[0164] La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden estar sumamente integradas, pero se ilustran por separado con fines conceptuales. La estimación del movimiento, realizada por la unidad de estimación de movimiento 42, es el proceso de generación de vectores de movimiento, que estiman el movimiento para los bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de una PU de un bloque de vídeo dentro de una trama o imagen de vídeo actual, con respecto a un bloque predictivo dentro de una trama de referencia (u otra unidad codificada), con respecto al bloque actual que se está codificando dentro de la trama actual (u otra unidad codificada). Un bloque predictivo es un bloque que se revela como

estrechamente coincidente con el bloque a codificar, en lo que respecta a la diferencia de píxeles, lo cual puede determinarse mediante la suma de diferencias absolutas (SAD), la suma de las diferencias al cuadrado (SSD) u otras métricas de diferencia. En algunos ejemplos, el codificador de vídeo 20 puede calcular valores para posiciones fraccionarias de píxeles de imágenes de referencia almacenadas en la memoria de imágenes de referencia 64. Por ejemplo, el codificador de vídeo 20 puede interpolar valores de posiciones de un cuarto de píxel, posiciones de un octavo de píxel u otras posiciones fraccionarias de píxel de la imagen de referencia. Por lo tanto, la unidad de estimación de movimiento 42 puede realizar una búsqueda de movimiento relativa a las posiciones de píxel completo y las posiciones de píxel fraccionario, y emitir un vector de movimiento con una precisión de píxel fraccionario.

[0165] La unidad de estimación de movimiento 42 calcula un vector de movimiento para una PU de un bloque de vídeo en un fragmento intercodificado, comparando la posición de la PU con la posición de un bloque predictivo de una imagen de referencia. La imagen de referencia puede seleccionarse a partir de una primera lista (lista 0) de imágenes de referencia o una segunda lista (lista 1) de imágenes de referencia, cada una de las cuales identifica una o más imágenes de referencia almacenadas en una memoria de imágenes de referencia 64. La unidad de estimación de movimiento 42 envía el vector de movimiento calculado a la unidad de codificación por entropía 56 y a la unidad de compensación de movimiento 44.

[0166] La compensación de movimiento, llevada a cabo por la unidad de compensación de movimiento 44, puede implicar capturar o generar el bloque predictivo basándose en el vector de movimiento determinado por la unidad de estimación de movimiento 42. De nuevo, la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden integrarse funcionalmente, en algunos ejemplos. Tras recibir el vector de movimiento para la PU del bloque de vídeo actual, la unidad de compensación de movimiento 44 puede localizar el bloque predictivo al que apunta el vector de movimiento en una de las listas de imágenes de referencia. El sumador 50 forma un bloque de vídeo residual restando los valores de píxel del bloque predictivo a los valores de píxel del bloque de vídeo actual que se está codificando, generando valores de diferencias de píxel, como se expone más adelante. En general, la unidad de estimación de movimiento 42 lleva a cabo la estimación de movimiento con respecto a los componentes de luma, y la unidad de compensación de movimiento 44 utiliza los vectores de movimiento calculados basándose en los componentes de luma, tanto para los componentes de croma como para los componentes de luma. La unidad de selección de modalidad 40 también puede generar elementos sintácticos asociados a los bloques de vídeo y al fragmento de vídeo para su uso mediante el decodificador de vídeo 30 en la decodificación de los bloques de vídeo del fragmento de vídeo.

[0167] La unidad de intrapredicción 46 puede intrapredecir un bloque actual, como alternativa a la interpredicción llevada a cabo por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44, como se ha descrito anteriormente. En particular, la unidad de intrapredicción 46 puede determinar una modalidad de intrapredicción a usar para codificar un bloque actual. En algunos ejemplos, la unidad de intrapredicción 46 puede codificar un bloque actual usando diversas modalidades de intrapredicción, por ejemplo, durante pases de codificación independientes, y la unidad de intrapredicción 46 (o la unidad de selección de modalidad 40, en algunos ejemplos) puede seleccionar una modalidad de intrapredicción adecuada para usar, entre las modalidades probadas.

[0168] Por ejemplo, la unidad de intrapredicción 46 puede calcular valores de velocidad-distorsión usando un análisis de velocidad-distorsión para las diversas modalidades de intrapredicción probadas, y seleccionar la modalidad de intrapredicción que tenga las mejores características de velocidad-distorsión entre las modalidades probadas. El análisis de velocidad-distorsión determina, en general, una cantidad de distorsión (o de errores) entre un bloque codificado y un bloque original, no codificado, que fue codificado para producir el bloque codificado, así como una velocidad de transmisión de bits (es decir, un número de bits) usada para producir el bloque codificado. La unidad de intrapredicción 46 puede calcular razones a partir de las distorsiones y velocidades para los diversos bloques codificados, para determinar qué modalidad de intrapredicción presenta el mejor valor de velocidad-distorsión para el bloque.

[0169] Además, la unidad de intrapredicción 46 puede configurarse para codificar bloques de profundidad de un mapa de profundidad. Por ejemplo, la unidad de intrapredicción 46 usa modalidades de intrapredicción a partir de la norma HEVC básica (bidimensional) (como se describe, por ejemplo, con respecto a la figura 2 anteriormente), modalidades de modelado de profundidad (DMM) (como se describe, por ejemplo, con respecto a las figuras 3A y 3B anteriormente) y codificación de cadenas de fronteras regionales (como se describe, por ejemplo, con respecto a la figura 10 más adelante), para codificar una PU intrapredicha de un fragmento de profundidad.

[0170] En cualquier caso, el codificador de vídeo 20 forma un bloque de vídeo residual restando los datos de predicción de la unidad de selección de modalidad 40 al bloque de vídeo original que está codificándose. El sumador 50 representa el componente o los componentes que realizan esta operación de resta. La unidad de procesamiento de transformación 52 aplica una transformación, tal como una transformación discreta de coseno (DCT) o una transformación conceptualmente similar, al bloque residual, produciendo un bloque de vídeo que comprende valores residuales de coeficientes de transformación. La unidad de procesamiento de transformación 52 puede llevar a cabo otras transformaciones que son conceptualmente similares a la DCT. También se podrían usar transformaciones de ondículas, transformaciones de números enteros, transformaciones de subbandas u otros tipos de transformaciones.

5 **[0171]** En cualquier caso, la unidad de procesamiento de transformación 52 aplica la transformación al bloque residual, produciendo un bloque de coeficientes de transformación residuales. La transformación puede convertir la información residual desde un dominio de valores de píxel a un dominio de transformación, tal como un dominio de frecuencia. La unidad de procesamiento de transformación 52 puede enviar los coeficientes de transformación resultantes a la unidad de cuantización 54. La unidad de cuantización 54 cuantiza los coeficientes de transformación para reducir más la velocidad de bits. El proceso de cuantización puede reducir la profundidad de bits asociada a algunos, o a la totalidad, de los coeficientes. El grado de cuantización se puede modificar ajustando un parámetro de cuantización. En algunos ejemplos, la unidad de cuantización 54 puede realizar, a continuación, una exploración de la matriz que incluye los coeficientes de transformación cuantizados. De forma alternativa, la unidad de codificación por entropía 56 puede realizar la exploración.

15 **[0172]** Tras la cuantización, la unidad de codificación por entropía 56 codifica por entropía los coeficientes de transformación cuantizados. Por ejemplo, la unidad de codificación por entropía 56 puede llevar a cabo la codificación de longitud variable adaptativa al contexto (CAVLC), la codificación aritmética binaria adaptativa el contexto (CABAC), la codificación aritmética binaria adaptativa el contexto basada en sintaxis (SBAC), la codificación por entropía por división en intervalos de probabilidad (PIPE) u otra técnica de codificación por entropía. En el caso de la codificación por entropía basada en el contexto, el contexto se puede basar en bloques vecinos. Tras la codificación por entropía por parte de la unidad de codificación por entropía 56, el flujo de bits codificado se puede transmitir a otro dispositivo (por ejemplo, el decodificador de vídeo 30) o archivar para su posterior transmisión o recuperación.

20 **[0173]** La unidad de cuantización inversa 58 y la unidad de transformación inversa 60 aplican la cuantización inversa y la transformación inversa, respectivamente, para reconstruir el bloque residual en el dominio de píxeles, por ejemplo, para su uso posterior como bloque de referencia. La unidad 44 de compensación de movimiento puede calcular un bloque de referencia añadiendo el bloque residual a un bloque predictivo de una de las tramas de la memoria de imágenes de referencia 64. La unidad de compensación de movimiento 44 también puede aplicar uno o más filtros de interpolación al bloque residual reconstruido para calcular valores fraccionarios de píxeles, para su uso en la estimación de movimiento. El sumador 62 añade el bloque residual reconstruido al bloque de predicción compensado por movimiento, producido por la unidad de compensación de movimiento 44, para producir un bloque de vídeo reconstruido para su almacenamiento en la memoria de imágenes de referencia 64. El bloque de vídeo reconstruido puede ser utilizado por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44, como bloque de referencia para intercodificar un bloque en una trama de vídeo posterior.

35 **[0174]** En este sentido, el codificador de vídeo 20 puede configurarse para implementar una o más técnicas ejemplares descritas en esta divulgación. Por ejemplo, la unidad de intrapredicción 46 puede configurarse para implementar las técnicas ejemplares de codificación por intrapredicción, para un bloque de profundidad descrito en esta divulgación. En algunos ejemplos, la unidad de intrapredicción 46 en combinación con otros procesadores puede configurarse para implementar las técnicas descritas en esta divulgación.

40 **[0175]** En algunos ejemplos, el codificador de vídeo 20 puede configurarse para determinar un patrón de partición a partir de patrones de partición para bloques de un primer tamaño (por ejemplo, determinar un patrón de partición asociado a un bloque de tamaño 16x16). El codificador de vídeo 20 puede determinar una partición para un bloque de profundidad de un segundo tamaño en función del patrón de partición determinado a partir de los patrones de partición para bloques del primer tamaño, donde el segundo tamaño es más grande que el primer tamaño (por ejemplo, determinar una partición para un bloque de profundidad de tamaño 32x32 basándose en el patrón de partición determinado asociado a un bloque de tamaño 16x16). El codificador de vídeo 20 puede codificar por intrapredicción el bloque de profundidad en función del patrón de partición determinado para el bloque de profundidad.

50 **[0176]** En este ejemplo, el bloque de tamaño 16x16 (por ejemplo, un bloque del primer tamaño) puede no ser un bloque real dentro de la imagen que incluye el bloque de profundidad. Más bien, el bloque de tamaño 16x16 es un bloque conceptual, donde los patrones de partición que se usan para un bloque de tamaño 16x16 se usan en cambio para un bloque de tamaño 32x32 (por ejemplo, bloque de profundidad del segundo tamaño). Sin embargo, en algunos ejemplos, el bloque de tamaño 16x16 puede ser un bloque real de la imagen, o un bloque dentro del bloque de tamaño 32x32.

55 **[0177]** El codificador de vídeo 20 puede configurarse para determinar un patrón de partición (por ejemplo, un patrón de partición basado en líneas) para un subbloque (por ejemplo, un bloque de tamaño MxM) de un bloque de profundidad (por ejemplo, un bloque de tamaño NxN que es una PU actual) para la intracodificación de profundidad. El codificador de vídeo 20 puede determinar un patrón de partición (por ejemplo, un patrón de partición basado en líneas) para todo el bloque de profundidad, en función del patrón de partición para el subbloque. El codificador de vídeo 20 puede intracodificar el bloque de profundidad en función del patrón de partición determinado para el bloque de profundidad.

65 **[0178]** El codificador de vídeo 20 puede codificar información que indica el patrón de partición para el subbloque. El codificador de vídeo 20 también puede evitar codificar (por ejemplo, no codificar) la información que indica el patrón de partición para el bloque de profundidad. El codificador de vídeo 20 también puede determinar el patrón de partición para el bloque de profundidad sin basar el patrón de partición para el bloque de profundidad en un tamaño del bloque

de profundidad (por ejemplo, independiente del tamaño del bloque). El codificador de vídeo 20 también puede, en algunos ejemplos, codificar (por ejemplo, usando la modalidad de omisión y/o el modelado de contexto) y señalar información para identificar el subbloque cuyo patrón de partición fue determinado (por ejemplo, un índice o un indicador y un desplazamiento).

5
10
15
[0179] La figura 7 es un diagrama de bloques que ilustra un ejemplo del decodificador de vídeo 30 que puede implementar técnicas para la codificación de profundidad. En el ejemplo de la figura 7, el decodificador de vídeo 30 incluye una memoria de datos de vídeo 69, una unidad de decodificación por entropía 70 y una unidad de procesamiento de predicción 71 que incluye una unidad de compensación de movimiento 72, una unidad de predicción de vectores de movimiento 73 y una unidad de intrapredicción 74. El decodificador de vídeo 30 también incluye la unidad de cuantización inversa 76, la unidad de procesamiento de transformación inversa 78, la memoria de tramas de referencia 82 y el sumador 80. En algunos ejemplos, el decodificador de vídeo 30 puede realizar un pase de decodificación, en general, recíproco al pase de codificación descrito con respecto al codificador de vídeo 20 (figura 6). La unidad de compensación de movimiento 72 puede generar datos de predicción basándose en vectores de movimiento recibidos desde la unidad de decodificación por entropía 70, mientras que la unidad de intrapredicción 74 puede generar datos de predicción basándose en indicadores de modalidad de intrapredicción, recibidos desde la unidad de decodificación por entropía 70.

20
25
[0180] En el ejemplo de la figura 7, la memoria de datos de vídeo 69 recibe vídeo codificado. La memoria de datos de vídeo 69 puede almacenar datos de vídeo (por ejemplo, configurados para almacenar datos de vídeo), tales como un flujo de bits de vídeo codificado, para ser decodificados por los componentes del decodificador de vídeo 30B. La memoria de datos de vídeo 68 también puede almacenar datos de vídeo que son utilizados por los componentes del decodificador de vídeo 30 para decodificar los datos de vídeo y reconstruir una imagen. Por ejemplo, la memoria de datos de vídeo 69 puede almacenar patrones de partición (por ejemplo, patrones de partición de minicuñas) para bloques de diferentes tamaños. La memoria de datos de vídeo 69 puede almacenar los patrones de partición como parte del proceso de inicialización implementado por el codificador de vídeo 30.

30
[0181] Los datos de vídeo almacenados en la memoria de datos de vídeo 69 pueden obtenerse desde un origen de vídeo local, tal como una cámara, mediante la comunicación de datos de vídeo de red cableada o inalámbrica, o al acceder a medios físicos de almacenamiento de datos. La memoria de datos de vídeo 69 puede formar una memoria intermedia de imágenes codificadas (CPB) que almacene datos de vídeo codificados a partir de un flujo de bits de vídeo codificado.

35
40
[0182] La memoria de imágenes de referencia 82 es un ejemplo de una memoria intermedia de imágenes decodificadas (DPB) que almacena datos de vídeo de referencia para su uso en la decodificación de datos de vídeo por parte del decodificador de vídeo 30 (por ejemplo, en la modalidad de intracodificación y la modalidad de intercodificación). La memoria de datos de vídeo 69 y la memoria de imágenes de referencia 82 pueden ser formadas por cualquiera entre varios dispositivos de memoria, tales como memoria dinámica de acceso aleatorio (DRAM), incluyendo DRAM síncrona (SDRAM), RAM magnetorresistiva (MRAM), RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. La memoria de datos de vídeo 69 y la memoria de imágenes de referencia 82 pueden ser proporcionadas por el mismo dispositivo de memoria o por dispositivos de memoria independientes. En diversos ejemplos, la memoria de datos de vídeo 69 puede estar en un chip con otros componentes del decodificador de vídeo 30, o fuera de chip con respecto a esos componentes.

45
50
[0183] Durante el proceso de decodificación, el decodificador de vídeo 30 recibe un flujo de bits de vídeo codificado, que representa bloques de vídeo de un fragmento de vídeo codificado y elementos sintácticos asociados, desde el codificador de vídeo 20. La unidad de decodificación por entropía 70 del decodificador de vídeo 30 decodifica por entropía el flujo de bits para generar coeficientes cuantizados, vectores de movimiento o indicadores de modalidad de intrapredicción y otros elementos sintácticos. La unidad de decodificación por entropía 70 remite los vectores de movimiento y otros elementos sintácticos a la unidad de compensación de movimiento 72. La unidad de decodificación por entropía 70 remite la información para la decodificación por intrapredicción a la unidad de intrapredicción 74. El decodificador de vídeo 30 puede recibir los elementos sintácticos en el nivel de fragmento de vídeo y/o el nivel de bloque de vídeo.

55
60
[0184] Como se ha indicado anteriormente, el decodificador de vídeo 30 puede adaptarse para realizar la codificación de vídeo de vistas múltiples. En algunos casos, el decodificador 30 de vídeo puede configurarse para decodificar la norma HEVC de vistas múltiples. Para la HEVC-3D, además de decodificar valores de textura (es decir, valores de luma y croma) para cada vista, el decodificador de vídeo 30 puede decodificar adicionalmente un mapa de profundidad para cada vista.

65
[0185] En cualquier caso, cuando el fragmento de vídeo se codifica como un fragmento intracodificado (I), la unidad de intrapredicción 74 puede generar datos de predicción para un bloque de vídeo del fragmento de vídeo actual, basándose en una modalidad de intrapredicción señalizada y datos de bloques previamente decodificados de la trama o imagen actual.

[0186] La unidad de intrapredicción 74 también puede intracodificar datos de profundidad. Por ejemplo, la unidad de

intrapredicción 74 usa modalidades de intrapredicción a partir de la norma HEVC básica (bidimensional) (como se describe, por ejemplo, con respecto a la figura 2 anteriormente), modalidades de modelado de profundidad (DMM) (como se describe, por ejemplo, con respecto a las figuras 3A y 3B a continuación) y codificación de cadenas de fronteras regionales (como se describe, por ejemplo, con respecto a la figura 10 a continuación) para codificar una PU intrapredicha de un fragmento de profundidad.

[0187] Cuando la trama de vídeo se codifica como un fragmento intercodificado (es decir, B (predicho bidireccionalmente), P (predicho a partir de una trama anterior) o GPB (fragmento P o B generalizado)), la unidad de compensación de movimiento 72 genera bloques predictivos para un bloque de vídeo del fragmento de vídeo actual basándose en los vectores de movimiento y en otros elementos sintácticos recibidos desde la unidad de decodificación por entropía 70. Los bloques predictivos se pueden generar a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El decodificador de vídeo 30 puede construir las listas de tramas de referencia, la Lista 0 y la Lista 1, usando técnicas de construcción por omisión, basándose en las imágenes de referencia almacenadas en la memoria de imágenes de referencia 82.

[0188] La unidad de compensación de movimiento 72 determina la información de predicción para un bloque de vídeo del fragmento de vídeo actual, analizando sintácticamente los vectores de movimiento y otros elementos sintácticos, y usa la información de predicción para generar los bloques predictivos para el bloque de vídeo actual que se está decodificando. Por ejemplo, la unidad de compensación de movimiento 72 usa algunos de los elementos sintácticos recibidos para determinar una modalidad de predicción (por ejemplo, intrapredicción o interpredicción) usada para codificar los bloques de vídeo del fragmento de vídeo, un tipo de fragmento de interpredicción (por ejemplo, un fragmento B, un fragmento P o un fragmento GPB), información de construcción para una o más de las listas de imágenes de referencia para el fragmento, vectores de movimiento para cada bloque de vídeo intercodificado del fragmento, el estado de interpredicción para cada bloque de vídeo intercodificado del fragmento y otra información para decodificar los bloques de vídeo en el fragmento de vídeo actual.

[0189] La unidad de compensación de movimiento 72 también puede realizar la interpolación basándose en filtros de interpolación. La unidad de compensación de movimiento 72 puede usar filtros de interpolación como los usados por el codificador de vídeo 20 durante la codificación de los bloques de vídeo, para calcular valores interpolados para píxeles fraccionarios de los bloques de referencia. En este caso, la unidad de compensación de movimiento 72 puede determinar los filtros de interpolación usados por el codificador de vídeo 20 a partir de los elementos sintácticos recibidos y usar los filtros de interpolación para generar bloques predictivos.

[0190] La unidad de cuantización inversa 76 cuantiza inversamente, es decir, decuantiza, los coeficientes de transformación cuantizados proporcionados en el flujo de bits y decodificados por la unidad de decodificación por entropía 80. El proceso de cuantización inversa puede incluir el uso de un parámetro de cuantización QPy, calculado por el decodificador de vídeo 30 para cada bloque de vídeo en el fragmento de vídeo, para determinar un grado de cuantización y, asimismo, un grado de cuantización inversa que debería aplicarse.

[0191] La unidad de procesamiento de transformación inversa 78 aplica una transformación inversa, por ejemplo una DCT inversa, una transformación inversa entera o un proceso de transformación inversa conceptualmente similar, a los coeficientes de transformación, con el fin de generar bloques residuales en el dominio del píxel. Después de que la unidad de compensación de movimiento 72, o la unidad de intrapredicción 74, genera el bloque predictivo para el bloque de vídeo actual (por ejemplo, un bloque de textura o un bloque de profundidad) basándose en vectores de movimiento u otros elementos sintácticos, el decodificador de vídeo 30 forma un bloque de vídeo decodificado sumando los bloques residuales de la unidad de transformación inversa 78 y los correspondientes bloques predictivos generados por la unidad de compensación de movimiento 82, o la unidad de intrapredicción 74. El sumador 90 representa el componente o los componentes que realizan esta operación de suma.

[0192] Si se desea, también se puede aplicar un filtro de eliminación de bloques para filtrar los bloques decodificados, a fin de eliminar distorsiones de efecto pixelado. También se pueden usar otros filtros de bucle (en el bucle de codificación o bien después del bucle de codificación) para allanar las transiciones de píxeles o mejorar de otro modo la calidad del vídeo. Los bloques de vídeo decodificados en una trama o imagen dada se almacenan a continuación en la memoria de imágenes de referencia 82, que almacena imágenes de referencia usadas para una compensación de movimiento subsiguiente. La memoria de imágenes de referencia 82 almacena también vídeo decodificado para su presentación posterior en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la figura 1.

[0193] En este sentido, el decodificador de vídeo 30 puede configurarse para implementar una o más técnicas ejemplares descritas en esta divulgación. Por ejemplo, la unidad de intrapredicción 72 puede configurarse para implementar las técnicas ejemplares de decodificación de intrapredicción para un bloque de profundidad descrito en esta divulgación. En algunos ejemplos, la unidad de intrapredicción 72, sola o en combinación con otros procesadores, puede configurarse para implementar las técnicas descritas en esta divulgación.

[0194] En algunos ejemplos, el decodificador de vídeo 30 puede configurarse para determinar un patrón de partición a partir de patrones de partición para bloques de un primer tamaño (por ejemplo, determinar un patrón de partición asociado a un bloque de tamaño 16x16). El decodificador de vídeo 30 puede determinar una partición para un bloque

de profundidad de un segundo tamaño basándose en el patrón de partición determinado a partir de los patrones de partición para bloques del primer tamaño, donde el segundo tamaño es más grande que el primer tamaño (por ejemplo, determinar una partición para un bloque de profundidad de tamaño 32x32 basándose en el patrón de partición determinado del bloque de tamaño 16x16). El decodificador de vídeo 30 puede decodificar por intrapredicción el bloque de profundidad basándose en el patrón de partición determinado para el bloque de profundidad.

[0195] En este ejemplo, el bloque de tamaño 16x16 (por ejemplo, un bloque del primer tamaño) puede no ser un bloque real dentro de la imagen que incluye el bloque de profundidad. Más bien, el bloque de tamaño 16x16 es un bloque conceptual, donde los patrones de partición que se usan para un bloque de tamaño 16x16 se usan en cambio para un bloque de tamaño 32x32 (por ejemplo, bloque de profundidad del segundo tamaño). Sin embargo, en algunos ejemplos, el bloque de tamaño 16x16 puede ser un bloque real de la imagen, o un bloque dentro del bloque de tamaño 32x32.

[0196] Por ejemplo, el decodificador de vídeo 30 puede configurarse para determinar un patrón de partición (por ejemplo, recibir información que indica el patrón de partición) para un subbloque (por ejemplo, un bloque de tamaño MxM) de un bloque de profundidad (por ejemplo, un bloque de tamaño NxN que es una PU actual) para intradecodificación de profundidad. El patrón de partición puede ser un patrón de partición basado en líneas. El decodificador de vídeo 30 puede determinar un patrón de partición (por ejemplo, un patrón de partición basado en líneas) para todo el bloque de profundidad basándose en el patrón de partición para el subbloque. El decodificador de vídeo 30 puede intradecodificar el bloque de profundidad en función del patrón de partición determinado para el bloque de profundidad.

[0197] El decodificador de vídeo 30 puede decodificar información que indica el patrón de partición para el subbloque. El decodificador de vídeo 30 también puede evitar decodificar (por ejemplo, no decodificar) la información que indica el patrón de partición para el bloque de profundidad. En otras palabras, el decodificador de vídeo 30 puede determinar el patrón de partición para el bloque de profundidad sin recibir información que indique el patrón de partición para el bloque de profundidad.

[0198] El decodificador de vídeo 30 también puede determinar el patrón de partición para el bloque de profundidad sin basar el patrón de partición para el bloque de profundidad en un tamaño del bloque de profundidad (por ejemplo, independiente del tamaño del bloque). El decodificador de vídeo 30 también puede, en algunos ejemplos, decodificar (por ejemplo, usando la modalidad de omisión y/o el modelado de contexto) información recibida para identificar el subbloque cuyo patrón de partición fue determinado (por ejemplo, un índice o un indicador y un desplazamiento).

[0199] Las técnicas descritas anteriormente pueden realizarse mediante el codificador de vídeo 20 (figuras 1 y 6) y/o el decodificador de vídeo 30 (figuras 1 y 7), de los cuales ambos pueden denominarse en general codificadores de vídeo. Además, la codificación de vídeo puede referirse en general a la codificación de vídeo y / o a la decodificación de vídeo, según corresponda.

[0200] La figura 8 es un diagrama de flujo que ilustra un procedimiento ejemplar de decodificación de datos de vídeo. Como se ilustra, el decodificador de vídeo 30 puede construir y almacenar patrones de una partición para bloques de un primer tamaño (800). Por ejemplo, el decodificador de vídeo 30 puede construir una lista de patrones de partición que incluya uno o más patrones de partición asociados a bloques de un primer tamaño (por ejemplo, para bloques de tamaño 16x16) como parte de la inicialización y almacenar la lista de patrones de partición asociada a bloques del primer tamaño en datos de memoria de vídeo 69.

[0201] El decodificador de vídeo 30 puede determinar un patrón de partición asociado a un bloque de un primer tamaño (802). Por ejemplo, el decodificador de vídeo 30 puede recibir un índice de la lista de patrones de partición, asociada a un bloque del primer tamaño. El decodificador de vídeo 30 puede identificar el patrón de partición al que hace referencia el índice de la lista de patrones de partición. El decodificador de vídeo 30 puede determinar el patrón de partición a partir de los patrones de partición asociados a los bloques del primer tamaño, en función del patrón de partición identificado (es decir, el decodificador de vídeo 30 puede determinar el patrón de partición a partir de los patrones de partición para bloques del primer tamaño, en función del índice recibido de la lista de patrones de partición).

[0202] El decodificador de vídeo 30 puede determinar un patrón de partición para un bloque de profundidad de un segundo tamaño (por ejemplo, 32x32) basándose en el patrón de partición determinado a partir de los patrones de partición para bloques del primer tamaño (por ejemplo, 16x16) (804). En este ejemplo, el segundo tamaño es más grande que el primer tamaño.

[0203] El bloque del primer tamaño no tiene por qué ser necesariamente un bloque real dentro de la imagen. Más bien, es el patrón de partición de un bloque del primer tamaño que el decodificador de vídeo 30 utiliza para determinar un patrón de partición para un bloque de profundidad de un segundo tamaño más grande. Sin embargo, en algunos ejemplos, el bloque del primer tamaño puede ser un bloque real dentro de la imagen, tal como dentro del bloque de profundidad.

- 5 **[0204]** En algunos ejemplos, el decodificador de vídeo 30 puede recibir información que identifica un subbloque dentro del bloque de profundidad, donde un tamaño del subbloque es el primer tamaño. En tales ejemplos, el decodificador de vídeo 30 puede determinar el patrón de partición para el subbloque. El decodificador de vídeo 30 también puede determinar el patrón de partición para el bloque de profundidad del segundo tamaño en función del patrón de partición determinado para el subbloque. Además, en estas técnicas ejemplares, el decodificador de vídeo 30 puede determinar el patrón de partición para el bloque de profundidad del segundo tamaño al extender una línea lineal del patrón de partición para el subbloque a las fronteras del bloque de profundidad. Una línea lineal resultante comprende el patrón de partición para el bloque de profundidad.
- 10 **[0205]** El decodificador de vídeo 30 puede decodificar por intrapredicción el bloque de profundidad basándose en el patrón de partición determinado para el bloque de profundidad (806). Por ejemplo, el decodificador de vídeo 30 puede implementar técnicas de decodificación por intrapredicción descritas en la norma 3D-HEVC, que se basan en las DMM para decodificar por intrapredicción el bloque de profundidad. En algunos ejemplos, el patrón de partición proveniente de los patrones de partición para bloques del primer tamaño comprende un patrón de partición basado en líneas, a partir de una pluralidad de patrones de partición basados en líneas, para bloques del primer tamaño (por ejemplo, un patrón de minicuñas a partir de una pluralidad de patrones de minicuñas para bloques del primer tamaño). Además, el patrón de partición para el bloque de profundidad comprende un patrón de partición basado en líneas para el bloque de profundidad que divide el bloque de profundidad en dos particiones. En algunos ejemplos, el decodificador de vídeo 30 puede determinar el patrón de partición para el bloque de profundidad del segundo tamaño sin recibir información que indique el patrón de partición para el bloque de profundidad del segundo tamaño.
- 15 **[0206]** La figura 9 es un diagrama de flujo que ilustra un procedimiento ejemplar de codificación de datos de vídeo. Como se ilustra, el codificador de vídeo 20 puede construir y almacenar patrones de una partición asociados a bloques de un primer tamaño (900). Por ejemplo, el codificador de vídeo 20 puede construir una lista de patrones de partición asociada a un bloque de un primer tamaño (por ejemplo, para un bloque de tamaño 16x16) como parte de la inicialización y almacenar la lista de patrones de partición para bloques del primer tamaño en los datos de memoria de vídeo 39.
- 20 **[0207]** El codificador de vídeo 20 puede determinar un patrón de partición asociado a un bloque del primer tamaño (902). Por ejemplo, el codificador de vídeo 20 puede implementar múltiples pases de codificación para seleccionar el patrón de partición adecuado que se puede usar para determinar un patrón de partición para un bloque de profundidad de un segundo tamaño más grande. El codificador de vídeo 20 puede identificar entonces un patrón de partición en la lista de patrones de partición. Por ejemplo, el codificador de vídeo 20 puede determinar el patrón de partición a partir del patrón de partición asociado a bloques del primer tamaño de un patrón de partición identificado en la lista de patrones de partición construida. El codificador de vídeo 20 puede señalar un índice de la lista de patrones de partición construida, que identifica el patrón de partición determinado a partir de los patrones de partición para bloques del primer tamaño, donde el decodificador de vídeo 30 usa el índice para decodificar por intrapredicción el bloque de profundidad.
- 25 **[0208]** El codificador de vídeo 20 puede determinar un patrón de partición para un bloque de profundidad de un segundo tamaño (por ejemplo, 32x32) basándose en el patrón de partición determinado a partir de los patrones de partición para bloques del primer tamaño (por ejemplo, un patrón de partición asociado a un bloque de tamaño 16x16) (904). En este ejemplo, el segundo tamaño es más grande que el primer tamaño.
- 30 **[0209]** El bloque del primer tamaño no tiene por qué ser necesariamente un bloque real dentro de la imagen. Más bien, es el patrón de partición de un bloque del primer tamaño que el codificador de vídeo 20 utiliza para determinar un patrón de partición para un bloque de profundidad de un segundo tamaño más grande. Sin embargo, en algunos ejemplos, el bloque del primer tamaño puede ser un bloque real dentro de la imagen, tal como dentro del bloque de profundidad.
- 35 **[0210]** En algunos ejemplos, el codificador de vídeo 20 puede identificar un subbloque dentro del bloque de profundidad, donde un tamaño del subbloque es el primer tamaño. En tales ejemplos, el codificador de vídeo 20 puede determinar el patrón de partición para el subbloque identificado. El codificador de vídeo 20 también puede determinar el patrón de partición para el bloque de profundidad del segundo tamaño en función del patrón de partición determinado para el subbloque identificado. Además, en estas técnicas ejemplares, el codificador de vídeo 20 puede determinar el patrón de partición para el bloque de profundidad del segundo tamaño al extender una línea lineal del patrón de partición para el subbloque a las fronteras del bloque de profundidad. Una línea lineal resultante define el patrón de partición para el bloque de profundidad.
- 40 **[0211]** El codificador de vídeo 20 puede codificar por intrapredicción el bloque de profundidad basándose en el patrón de partición determinado para el bloque de profundidad (906). Por ejemplo, el codificador de vídeo 20 puede implementar técnicas de codificación por intrapredicción descritas en la norma 3D-HEVC, que se apoyan en las DMM para codificar por intrapredicción el bloque de profundidad. En algunos ejemplos, el patrón de partición proveniente de los patrones de partición para bloques del primer tamaño comprende un patrón de partición basado en líneas, a partir de una pluralidad de patrones de partición basados en líneas, para bloques del primer tamaño (por ejemplo, un patrón de minicuñas a partir de una pluralidad de patrones de minicuñas para bloques del primer tamaño). Además, el patrón
- 45 **[0209]** El bloque del primer tamaño no tiene por qué ser necesariamente un bloque real dentro de la imagen. Más bien, es el patrón de partición de un bloque del primer tamaño que el codificador de vídeo 20 utiliza para determinar un patrón de partición para un bloque de profundidad de un segundo tamaño más grande. Sin embargo, en algunos ejemplos, el bloque del primer tamaño puede ser un bloque real dentro de la imagen, tal como dentro del bloque de profundidad.
- 50 **[0210]** En algunos ejemplos, el codificador de vídeo 20 puede identificar un subbloque dentro del bloque de profundidad, donde un tamaño del subbloque es el primer tamaño. En tales ejemplos, el codificador de vídeo 20 puede determinar el patrón de partición para el subbloque identificado. El codificador de vídeo 20 también puede determinar el patrón de partición para el bloque de profundidad del segundo tamaño en función del patrón de partición determinado para el subbloque identificado. Además, en estas técnicas ejemplares, el codificador de vídeo 20 puede determinar el patrón de partición para el bloque de profundidad del segundo tamaño al extender una línea lineal del patrón de partición para el subbloque a las fronteras del bloque de profundidad. Una línea lineal resultante define el patrón de partición para el bloque de profundidad.
- 55 **[0211]** El codificador de vídeo 20 puede codificar por intrapredicción el bloque de profundidad basándose en el patrón de partición determinado para el bloque de profundidad (906). Por ejemplo, el codificador de vídeo 20 puede implementar técnicas de codificación por intrapredicción descritas en la norma 3D-HEVC, que se apoyan en las DMM para codificar por intrapredicción el bloque de profundidad. En algunos ejemplos, el patrón de partición proveniente de los patrones de partición para bloques del primer tamaño comprende un patrón de partición basado en líneas, a partir de una pluralidad de patrones de partición basados en líneas, para bloques del primer tamaño (por ejemplo, un patrón de minicuñas a partir de una pluralidad de patrones de minicuñas para bloques del primer tamaño). Además, el patrón
- 60 **[0211]** El codificador de vídeo 20 puede codificar por intrapredicción el bloque de profundidad basándose en el patrón de partición determinado para el bloque de profundidad (906). Por ejemplo, el codificador de vídeo 20 puede implementar técnicas de codificación por intrapredicción descritas en la norma 3D-HEVC, que se apoyan en las DMM para codificar por intrapredicción el bloque de profundidad. En algunos ejemplos, el patrón de partición proveniente de los patrones de partición para bloques del primer tamaño comprende un patrón de partición basado en líneas, a partir de una pluralidad de patrones de partición basados en líneas, para bloques del primer tamaño (por ejemplo, un patrón de minicuñas a partir de una pluralidad de patrones de minicuñas para bloques del primer tamaño). Además, el patrón
- 65 **[0211]** El codificador de vídeo 20 puede codificar por intrapredicción el bloque de profundidad basándose en el patrón de partición determinado para el bloque de profundidad (906). Por ejemplo, el codificador de vídeo 20 puede implementar técnicas de codificación por intrapredicción descritas en la norma 3D-HEVC, que se apoyan en las DMM para codificar por intrapredicción el bloque de profundidad. En algunos ejemplos, el patrón de partición proveniente de los patrones de partición para bloques del primer tamaño comprende un patrón de partición basado en líneas, a partir de una pluralidad de patrones de partición basados en líneas, para bloques del primer tamaño (por ejemplo, un patrón de minicuñas a partir de una pluralidad de patrones de minicuñas para bloques del primer tamaño). Además, el patrón

de partición para el bloque de profundidad comprende un patrón de partición basado en líneas para el bloque de profundidad que divide el bloque de profundidad en dos particiones. En algunos ejemplos, el codificador de vídeo 20 puede no señalar información que identifique el patrón de partición para el bloque de profundidad del segundo tamaño.

5 **[0212]** A continuación se describen algunos ejemplos para implementar las técnicas ejemplares descritas en esta divulgación. En lo que sigue, los corchetes dobles ([[]]) que abarcan texto en negrita y en cursiva indican la eliminación en relación con las técnicas actuales, y el subrayado indica la adición en relación con las técnicas actuales. Se pueden aplicar las siguientes técnicas ejemplares, por separado o en combinación.

10

Para la sintaxis de parámetros de la modalidad de profundidad

depth_mode_parameters (x0, y0, log2CbSize) {	Descriptor
if ((log2CbSize == 3 && PartMode[xC][yC] == PART_2Nx2N) (log2CbSize > 3 && log2CbSize < 6))	
depth_intra_mode_set_indication_flag	ae(v)
depth_intra_mode[x0][y0]	ae(v)
if (DepthIntraMode[x0][y0] == INTRA_DEP_DMM_WFULL DepthIntraMode[x0][y0] == INTRA_DEP_SDC_DMM_WFULL) {	
wedge_full_tab_idx[x0][y0]	ae(v)
<u>wedge_sub_col[x0][y0]</u>	<u>u(v)</u>
<u>wedge_sub_row[x0][y0]</u>	<u>u(v)</u>
}	
else if (DepthIntraMode[x0][y0] == INTRA_DEP_DMM_WPREDTEX) {	
wedge_predtex_tab_idx[x0][y0]	<i>[[ae(v)]]ae(4)</i>
<u>wedge_sub_col[x0][y0]</u>	<u>u(v)</u>
<u>wedge_sub_row[x0][y0]</u>	<u>u(v)</u>
} else if (DepthIntraMode[x0][y0] == INTRA_DEP_CHAIN) {	
edge_start_left_flag[x0][y0]	ae(v)
edge_start_position[x0][y0]	ae(v)
num_edge_codes_minus1[x0][y0]	ae(v)
for(k = 0; k <= num_edge_codes_minus1; k++)	
edge_code[k]	ae(v)
}	
...	
}	

15

[0213] Para el parámetro de modalidad de Profundidad, intrasemántica:

wedge_full_tab_idx[x0][y0] especifica el índice del patrón de minicuñas en la correspondiente lista de patrones de tamaño 4x4 cuando DepthIntraMode[x0][y0] es igual a INTRA_DEP_DMM_WFULL.

20

wedge_sub_col[x0][y0] y wedge_sub_row[x0][y0] especifican la posición del bloque de tamaño 4x4 para el que se señala el patrón de minicuñas.

wedge_predtex_tab_idx[x0][y0] especifica el índice del patrón de minicuñas del bloque de tamaño 4x4 dentro de la unidad de predicción actual cuando DepthIntraMode[x0][y0] es igual a INTRA_PROF_DMM_WPREDTEX.

25

Para el proceso de decodificación

H.8.4.4.2.7 Especificación de modalidad de intrapredicción INTRA_DMM_WFULL

30

[0214] Las entradas para este proceso son:

- una ubicación de muestra (xTb, yTb) que especifica la muestra superior izquierda del bloque actual en relación con la muestra superior izquierda de la imagen actual,
- las muestras vecinas p[x][y], con $x = -1, y = -1 \dots nTbS * 2 - 1$ y $x = 0 \dots nTbS * 2 - 1, y = -1,$
- una variable nTbS que especifica el tamaño de bloque de la transformación,

[0215] La salida de este proceso es:

- las muestras predichas predSamples[x][y], con $x, y = 0 \dots nTbS - 1.$

[0216] Los valores de las muestras de predicción predSamples[x][y], con $x, y = 0 \dots nTbS - 1,$ se obtienen según lo especificado en las siguientes etapas ordenadas:

1. El proceso de extensión del patrón de minicuñas, como se especifica en la subcláusula H.8.4.2.4.12.3, se invoca con `Log2(nTbS)`, `wedge_full_tab_idx[xTb][yTb]`, `wedge_sub_col[xTb][yTb]` y `wedge_sub_row[xTb][yTb]` como entradas, y la salida se asigna a `wedgePattern`. **[La variable `wedgePattern[x][y]` con $x, y = 0 \dots nTbS - 1,$ especificando un patrón de partición binario, se obtiene como**

```
wedgePattern
= WedgePatternTable[ Log2( nTbS ) ][ wedge_full_tab_idx[ xTb ][ yTb ] ] (H-39))
```

]]

2. El proceso de obtención y asignación del valor de la partición de profundidad, como se especifica en la subcláusula H.8.4.2.4.11, se invoca con las muestras adyacentes p[x][y], el patrón binario `wedgePattern[xTb][yTb]`, el tamaño de transformación nTbS, el `dcOffsetAvailFlag` fijado igual a `dmm_dc_flag[xTb][yTb]`, el `intraChainFlag` fijado igual a 0 y los Desplazamientos de DC `DcOffset[xTb][yTb][0]` y `DcOffset[xTb][yTb][1]` como entradas, y la salida es asignada a `predSamples[x][y]`.

3. **[Para $x, y = 0 \dots nTbS - 1,$ inclusive, vale lo siguiente:**

- **`WedgeIdx[xTb + x][yTb + y]` se fija igual a `wedge_full_tab_idx[xTb][yTb]`.]**

H.8.4.4.2.8 Especificación de modalidad de intrapredicción INTRA_DMM_WPRETEX

[0217] Las entradas para este proceso son:

- una ubicación de muestra (xTb, yTb) que especifica la muestra superior izquierda del bloque actual en relación con la muestra superior izquierda de la imagen actual,
- las muestras vecinas p[x][y], con $x = -1, y = -1 \dots nTbS * 2 - 1$ y $x = 0 \dots nTbS * 2 - 1, y = -1.$
- una variable nTbS que especifica el tamaño de bloque de la transformación,

[0218] La salida de este proceso es:

- las muestras predichas predSamples[x][y], con $x, y = 0 \dots nTbS - 1.$

[0219] Los valores de las muestras de predicción predSamples[x][y], con $x, y = 0 \dots nTbS - 1,$ son los siguientes:

- Según la `textureIntraPredMode`, la variable `WedgeIdx` se obtiene como se especifica en lo que sigue:

```
candWedgeIndList =
TextModePredWedgeIndTable[[Log2(nTbS)]]2 [ TextureIntraPredMode[ xTb ][ yTb ] ] (H-40)
```

```
WedgeIdx = candWedgeIndList[ wedge_predtex_tab_idx[ xTb ][ yTb ] ] (H-41)
```

- El proceso de extensión del patrón de minicuñas, como se especifica en la subcláusula H.8.4.2.4.12.3, se invoca con `Log2(nTbS)`, `WedgeIdx`, `wedge_sub_col[xTb][yTb]` y `wedge_sub_row[xTb][yTb]` como entradas, y la salida se asigna a `wedgePattern`. **[La variable `wedgePattern[x][y]` con $x, y = 0 \dots nTbS - 1,$ que especifica un patrón de partición binario, se obtiene como**

```
wedgePattern = WedgePatternTable[ Log2( nTbS ) ][ WedgeIdx ] (H-42))
```

]]

- El proceso de obtención y asignación del valor de la partición de profundidad, como se especifica en la subcláusula H.8.4.2.4.11, se invoca con las muestras adyacentes $p[x][y]$, el patrón binario $wedgePattern[x][y]$, el tamaño de transformación nT , el $dcOffsetAvailFlag$ fijado igual a $dmm_dc_flag[xTb][yTb]$, el $intraChainFlag$ fijado igual a 0 y los Desplazamientos de DC $DcOffset[xTb][yTb][0]$ y $DcOffset[xTb][yTb][1]$ como entradas, y la salida es asignada a $predSamples[x][y]$.

H.8.4.2.4.12.3 Proceso de extensión de patrón de minicuñas

10 **[0220]** Las entradas para este proceso son:

- una variable $log2BlkSize$ que especifica el tamaño del patrón de partición binario
- una variable $subIdx$ que especifica un índice de minicuñas de tamaño 4×4 ,
- 15 - una variable $xSub$ que especifica la posición horizontal de un bloque superior izquierdo de tamaño 4×4 ,
- una variable $ySub$ que especifica la posición vertical de un bloque superior izquierdo de tamaño 4×4 ,

20 **[0221]** La salida de este proceso es:

- formación binaria $wedgePattern[x][y]$ de tamaño $(1 < \log2BlkSize) \times (1 < \log2BlkSize)$

25 **[0222]** Las variables xS e yS , que especifican la posición de inicio de la línea de partición de un patrón de minicuñas de tamaño 4×4 , se obtienen de la siguiente manera:

- $xS = WedgeStartPos[2][subIdx][0]$;
- $yS = WedgeStartPos[2][subIdx][1]$;

30 **[0223]** Las variables xE e yE , que especifican la posición final de la línea de partición de un patrón de minicuñas de tamaño 4×4 , se obtienen de la siguiente manera:

- $xE = WedgeEndPos[2][subIdx][0]$;
- $yE = WedgeEndPos[2][subIdx][1]$;

[0224] Si $log2BlkSize$ es igual a 2, vale lo siguiente:

- 40 - $WedgePatternTable[2][subIdx]$ se asigna a $wedgePattern$;

De lo contrario, vale lo siguiente:

- 45 - Si $xS \neq xE$ o $yS \neq yE$, vale lo siguiente:

- $a = (yE - yS)$;
- $b = (yE + (ySub \ll 3)) * (xE - xS) - a * (xE + (xSub \ll 3))$;

- 50 - Para $x, y = 0 \dots blkSize - 1$

$$- \quad \underline{wedgePattern[x][y] = ((y * (xE - xS) - a * x) \ll 1)}$$

- 55 - De lo contrario, vale lo siguiente:

- Para $x, y = 0 \dots blkSize - 1$

- Si $ySub = 0$, vale lo siguiente:

$$- \quad \underline{wedgePattern[x][y] = x < (xSub \ll 2) ? 1 : 0}$$

- De lo contrario, vale lo siguiente:

$$\underline{wedgePattern[x][y] = y < (ySub \ll 2) ? 1 : 0}$$

H.8.4.4.3 Proceso de reconstrucción del valor de profundidad

[0225] Las entradas para este proceso son:

- 5 - una ubicación de luma (xTb, yTb) que especifica la muestra superior izquierda de luma del bloque de codificación actual en relación con la muestra de luma superior izquierda de la imagen actual,
- una variable nTbS que especifica el tamaño de bloque de la transformación,
- 10 - muestras predichas predSamples[x][y], con x, y = 0...nTbS-1.
- la modalidad de intrapredicción predModelIntra,

[0226] La salida de este proceso es:

- 15 - muestras de valor de profundidad reconstruidas resSamples[x][y], con x, y = 0... nTbS - 1.

[0227] En función de predModelIntra, la formación wedgePattern[x][y] con x, y = 0...nTbS - 1, que especifica el patrón de segmentación binario, se obtiene de la siguiente manera.

- 20 - Si predModelIntra es igual a INTRA_DMM_WFULL, el proceso de extensión del patrón de minicuñas, como se especifica en la subcláusula H.8.4.2.4.12.3, se invoca con Log2(nTbS), el wedge_full_tab_idx[xTb][yTb], la wedge_sub_col[xTb][yTb] y la wedge_sub_row[xTb][yTb] como entradas, y la salida se asigna a wedgePattern. [[vale lo siguiente.

$$25 \quad \begin{aligned} & \mathbf{wedgePattern} \\ & = \mathbf{WedgePatternTable[Log2(nTbS)][wedge_full_tab_idx[xTb][yTb]]} \\ & \mathbf{]} \end{aligned}$$

- 30 - De lo contrario (predModelIntra no es igual a INTRA_DMM_WFULL), vale lo siguiente.
 - Para x, y = 0...nTbS - 1, wedgePattern[x][y] se fija igual a 0.

[0228] En función de dlt_flag[nuh_layer_id], las muestras de valor de profundidad reconstruidas resSamples[x][y] se obtienen como se especifica a continuación:

- 35 - Si dlt_flag[nuh_layer_id] es igual a 0, vale lo siguiente:
 - 40 - Para x, y = 0...nTbS - 1, las muestras de valor de profundidad reconstruidas resSamples[x][y] se obtienen como se especifica a continuación:

$$\text{resSamples[x][y]} = \text{predSamples[x][y]} + \text{SdcResidual[xTb][yTb][wedgePattern[x][y]]} \quad (\text{H-59})$$

- 45 - En otro caso (dlt_flag[nuh_layer_id] es igual a 1), vale lo siguiente:
 - Las variables dcPred[0] y dcPred[1] se obtienen como se especifica a continuación:
 - 50 - Si predModelIntra es igual a INTRA_DC, vale lo siguiente:

$$\text{dcPred[0]} = \text{predSamples[nTbS - 1][nTbS - 1]} \quad (\text{H-60})$$

- En caso contrario, si predModelIntra es igual a INTRA_PLANAR, vale lo siguiente:

$$55 \quad \text{dcPred[0]} = (\text{predSamples[0][0]} + \text{predSamples[0][nTbS - 1]} + \text{predSamples[nTbS - 1][0]} + \text{predSamples[nTbS - 1][nTbS - 1]} + 2) \gg 2 \quad (\text{H-61})$$

- Cuando (predModelIntra es igual a INTRA_DMM_WFULL), vale lo siguiente.

$$60 \quad \text{dcPred[wedgePattern[0][0]]} = \text{predSamples[0][0]} \quad (\text{H-62})$$

$$\text{dcPred[wedgePattern[nTbS - 1][0]]} = \text{predSamples[nTbS - 1][0]} \quad (\text{H-63})$$

$$\text{dcPred[wedgePattern[0][nTbS - 1]]} = \text{predSamples[0][nTbS - 1]} \quad (\text{H-64})$$

$$65 \quad \text{dcPred[wedgePattern[nTbS - 1][nTbS - 1]]} = \text{predSamples[nTbS - 1][nTbS - 1]} \quad (\text{H-65})$$

- Para $x, y = 0 \dots nTbS - 1$, las muestras de valores de profundidad reconstruidas $resSamples[x][y]$ se obtienen como se especifica a continuación:

$$dltIdxPred = DepthValue2Idx[dcPred[wedgePattern[x][y]]] \quad (H-66)$$

$$dltIdxResi = SdcResidual[xTb][yTb][wedgePattern[x][y]] \quad (H-67)$$

$$resSamples[x][y] = predSamples[x][y] + ldx2DepthValue[dltIdxPred + dltIdxResi] - dcPred[wedgePattern[x][y]] \quad (H-68)$$

[0229] En una segunda técnica ejemplar, similar a la primera técnica ejemplar, pero también se incluyen los siguientes cambios en el proceso de decodificación de la norma 3D-HEVC actual. Esta segunda técnica ejemplar se describe para determinar a qué partición pertenece una posición (m, n) en una PU de tamaño $N \times N$ con un patrón de minicuñas de tamaño $N \times N$, según a cuál partición pertenece la posición de decisión $(2m, 2n)$ en una PU de tamaño $2N \times 2N$ con el correspondiente patrón de minicuñas, y el desplazamiento de la posición de decisión $(2m, 2n)$ en el bloque de tamaño $2N \times 2N$, es decir, un vector de desplazamiento de $(desplazamientoX, desplazamientoY)$ agregado a $(2m, 2n)$, en donde $desplazamientoX$ o $desplazamientoY$ es igual a 0 o 1 y, según diversas condiciones como en la norma 3D-HEVC actual, se elimina. En algunos ejemplos, este cambio puede realizarse conjuntamente con las técnicas descritas anteriormente con respecto al ejemplo uno. En otros ejemplos, este cambio también puede hacerse independiente de las técnicas descritas en la primera técnica ejemplar.

H.8.4.2.4.12.1 Proceso de generación de patrones de minicuñas ...

[0230] 4. El patrón de partición binaria $wedgePattern[x][y]$, con $x, y = 0 \dots patternSize - 1$, se obtiene como se especifica a continuación.

- Si $resShift$ es igual a 1, vale lo siguiente:

[0230] - **[[En función de $wedgeOri$, las variables $xOff$ e $yOff$ se fijan como se especifica en la Tabla H-8.**

Tabla H-8 Especificación de $xOff$, $yOff$

<i>wedgeOri</i>	$(xS + xE) < curSize$	<i>xOff</i>	<i>yOff</i>
0		0	0
1		1	0
2		1	1
3		0	1
4	0	1	0
4	1	0	0
5	0	0	1
5	1	0	0

35 -]]

- Para $x, y = 0 \dots patternSize - 1$, vale lo siguiente:

$$wedgePattern[x][y] = curPattern[(x << 1) + [[xOff]]][(y << 1) + [[yOff]]] \quad (H-58)$$

- De lo contrario ($resShift$ no es igual a 1), $wedgePattern$ se fija igual a $curPattern$.

[0231] De manera similar a la primera técnica ejemplar, pero los siguientes cambios (indicados con subrayado y negrita, con respecto a la primera técnica ejemplar) también se incluyen para cubrir la partición de una muestra (la muestra superior izquierda en el patrón de partición pertenece a una partición diferente a todas las demás muestras) en el procedimiento propuesto para patrones de minicuñas de tamaño $N \times N$ ($N > 4$), y tal partición de una muestra puede no ser capaz de ser generada usando la primera técnica ejemplar.

H.8.4.2.4.12.3 Proceso de extensión de patrón de minicuñas

50 **[0232]** Las entradas para este proceso son:

- una variable log2BlkSize que especifica el tamaño del patrón de partición binaria
- una variable subIdx que especifica un índice de minicuñas de tamaño 4x4.
- 5 - una variable xSub que especifica la posición horizontal de un bloque superior izquierdo de tamaño 4x4.
- una variable ysub que especifica la posición vertical de un bloque superior izquierdo de tamaño 4x4.

[0233] La salida de este proceso es:

- formación binaria wedgePattern[x][y] de tamaño $(1 < \log_2 \text{BlkSize}) \times (1 < \log_2 \text{BlkSize})$

[0234] Las variables xS e yS, que especifican la posición de inicio de la línea de partición de un patrón de minicuñas de tamaño 4x4, se obtienen de la siguiente manera:

- $xS = \text{WedgeStartPos}[2][\text{subIdx}][0]$;
- $yS = \text{WedgeStartPos}[2][\text{subIdx}][1]$;

[0235] Las variables xE e yE, que especifican la posición final de la línea de partición de un patrón de minicuñas de tamaño 4x4, se obtienen de la siguiente manera:

- $xE = \text{WedgeEndPos}[2][\text{subIdx}][0]$;
- $yE = \text{WedgeEndPos}[2][\text{subIdx}][1]$;

[0236] Si log2BlkSize es igual a 2, vale lo siguiente:

- WedgePatternTable[2][subIdx] se asigna a wedgePattern; de lo contrario, vale lo siguiente:

- Si $xS \neq xE$ o $yS \neq yE$, vale lo siguiente:

- $a = (yE - yS)$;
- $b = (yE + (ySub \ll 3)) * (xE - xS) - a * (xE + (xSub \ll 3))$;
- Para $x, y = 0 \dots \text{blkSize} - 1$
- $\text{wedgePattern}[x][y] == ((y * (xE - xS) - a * x) \ll 1) < b ? 1 : 0$

- De lo contrario, vale lo siguiente:

- Para $x, y = 0 \dots \text{blkSize} - 1$
- **Si $xSub = 0$ && $ySub = 0$, vale lo siguiente:**

- **$\text{wedgePattern}[x][y] = (!x) \&\& (!y) ? 1 : 0$**

- **De lo contrario, si $ySub = 0$, vale lo siguiente:**

- $\text{wedgePattern}[x][y] = x < (xSub \ll 2) ? 1 : 0$

- De lo contrario, vale lo siguiente:

$\text{wedgePattern}[x][y] = y < (ySub \ll 2) ? 1 : 0$

[0237] Si bien las técnicas de esta divulgación se describen generalmente con respecto a la norma 3D-HEVC, las técnicas no están limitadas de esta manera. Las técnicas descritas anteriormente también pueden ser aplicables a otras normas actuales o normas futuras aún no desarrolladas. Por ejemplo, las técnicas para la codificación de profundidad también pueden ser aplicables a una extensión de múltiples vistas de la HEVC (por ejemplo, la llamada MV-HEVC), una extensión ajustable a escala de la HEVC u otras normas actuales o futuras que tengan un componente de profundidad.

[0238] A continuación se describen algunas técnicas adicionales relacionadas con las modalidades de intrapredicción en la norma 3D-HEVC. Esta divulgación se proporciona para asistir adicionalmente en la comprensión de las técnicas descritas en esta divulgación. Sin embargo, tales técnicas no son necesarias para esta divulgación, y

pueden utilizarse junto con las técnicas descritas en esta divulgación o pueden ser otro ejemplo de las técnicas descritas en esta divulgación.

[0239] Para la modalidad de codificación de cadenas de fronteras regionales, en la norma 3D-HEVC, la modalidad de codificación de cadenas de fronteras regionales se introduce junto con las modalidades de intrapredicción de la norma HEVC y las modalidades DMM, para codificar una unidad de intrapredicción de un fragmento de profundidad. Por brevedad, la "modalidad de codificación de cadenas de fronteras regionales" se indica por "codificación de cadenas", por simplicidad en los textos, tablas y figuras descritos en el presente documento.

[0240] El codificador de vídeo 20 puede señalar una codificación de cadena de una PU con una posición inicial de la cadena, el número de los códigos de cadena y el código de cadena para cada uno, y un índice de dirección. En general, una cadena es una conexión entre una muestra y una de sus ocho muestras de conectividad. Como se muestra en la figura 10, hay ocho tipos diferentes de cadena, cada uno asignado con un índice de dirección que va de 0 a 7.

[0241] La figura 10 es un diagrama conceptual que ilustra una modalidad de codificación de cadenas de fronteras regionales. Por ejemplo, la parte superior de la figura 10 ilustra ocho tipos posibles de cadenas definidas en la codificación de cadenas. La parte inferior de la figura 10 ilustra un patrón de partición de PU de profundidad y las cadenas codificadas en la codificación de cadenas.

[0242] Un ejemplo del proceso de codificación de cadenas se ilustra en la figura 10. Para señalar el patrón de partición arbitrario mostrado en la figura 10, el codificador de vídeo 20 puede identificar el patrón de partición y codifica la siguiente información en el flujo de bits. El codificador de vídeo 20 puede codificar un bit "0" para señalar que las cadenas comienzan desde la frontera superior. El codificador de vídeo 20 puede codificar tres bits "011" para señalar la posición inicial "3" en la frontera superior. El codificador de vídeo 20 puede codificar cuatro bits "0110" para señalar el número total de cadenas como 7. El codificador de vídeo 30 puede codificar una serie de índices de cadenas conectadas "3, 3, 3, 7, 1, 1, 1", donde cada índice de cadena se convierte en una palabra de código utilizando la tabla de consulta.

[0243] Por ejemplo, la norma 3D-HEVC incluye una modalidad de codificación de cadenas de fronteras regionales que permite la señalización explícita de las fronteras de partición (por ejemplo, en lugar de dividir basándose en un componente de textura cosituada, como se ha descrito anteriormente con respecto a las DMM). Como se ha señalado anteriormente, esta divulgación puede referirse a la "modalidad de codificación de cadenas de fronteras regionales" como "codificación de cadenas".

[0244] En general, una cadena es una conexión entre una muestra y una de sus ocho muestras de conectividad. Según lo mostrado por el bloque 160 de la figura 10, hay ocho tipos diferentes de dirección de cadena, cada uno asignado con un índice de dirección que va de 0 a 7. Los tipos de dirección de cadena pueden ayudar a un codificador de vídeo en la determinación de particiones de un bloque de profundidad.

[0245] Por ejemplo, el ejemplo de la figura 10 incluye una primera partición 162 y una segunda partición 164 separadas por una cadena 166 que indica la estructura de partición. Un codificador de vídeo (tal como el codificador de vídeo 20) puede determinar y señalar la cadena 166 para una PU en un flujo de bits codificado, mientras que un decodificador de vídeo (tal como el decodificador de vídeo 30) puede analizar sintácticamente datos que representan la cadena 166 proveniente de un flujo de bits codificado.

[0246] En general, la cadena 166 incluye una posición de inicio, una indicación de una serie de enlaces en la cadena (por ejemplo, un número de códigos de cadena) y, para cada código de cadena, un índice de dirección. También podrían usarse otros tipos de señalización para la cadena 166. En un ejemplo, para señalar el patrón de partición arbitrario que se muestra en el ejemplo de la figura 10, el codificador de vídeo 20 puede codificar un bit (por ejemplo, 0) para indicar que la cadena 166 comienza desde la frontera superior. El codificador de vídeo 20 puede codificar tres bits (por ejemplo, 011) para indicar que la cadena 166 comienza después de la tercera muestra de profundidad de la frontera superior. El codificador de vídeo 20 puede codificar cuatro bits (por ejemplo, 0110) para indicar que hay 7 enlaces totales en la cadena 166. El codificador de vídeo 20 también puede codificar una serie de índices de cadenas conectadas (por ejemplo, 3, 3, 3, 7, 1, 1, 1) para indicar la dirección de cada enlace de cadena (por ejemplo, de acuerdo al bloque 160). En algunos ejemplos, el codificador de vídeo 20 puede convertir cada índice en una palabra de código usando una tabla de consulta. Un decodificador de vídeo, tal como el decodificador de vídeo 30, puede analizar sintácticamente la señalización descrita anteriormente para determinar el patrón de partición de un bloque. El decodificador de vídeo 30 puede decodificar los valores de profundidad para cada partición.

[0247] Las técnicas descritas en esta divulgación también pueden funcionar junto con la codificación de profundidad simplificada (SDC), pero las técnicas no están limitadas de ese modo y pueden funcionar en ejemplos en los que no se utiliza la SDC. A continuación se describe la modalidad de codificación de profundidad simplificada (SDC). La modalidad de SDC se introduce junto con las modalidades de intrapredicción de la HEVC, las modalidades DMM y la modalidad de codificación de cadenas para codificar una intraPU de un fragmento de profundidad. En la norma 3D-HEVC actual, la SDC solo se aplica para el tamaño de partición $2N \times 2NPU$. En lugar de codificar los coeficientes de

transformación cuantizados, las modalidades de SDC representan un bloque de profundidad con los siguientes dos tipos de información: (1) el tipo de partición del bloque de profundidad actual, incluida la modalidad DMM 1 (2 particiones) y Plana (1 partición), y (2) para cada partición, se señala un valor residual (en el dominio de píxeles) en el flujo de bits.

5 **[0248]** Se definen dos sub-modalidades en la SDC, incluyendo la modalidad SDC 1 y la modalidad SDC 2, que corresponden al tipo de partición de modalidad Plana y DMM 1, respectivamente. La codificación residual simplificada se utiliza en la SDC. En la codificación residual simplificada, se señala un valor residual de DC para cada partición de la PU, y no se aplica ninguna transformación o cuantización.

10 **[0249]** Debería entenderse que, según el ejemplo, ciertos actos o sucesos de cualquiera de los procedimientos descritos en el presente documento pueden realizarse en una secuencia distinta, pueden añadirse, fundirse u omitirse por completo (por ejemplo, no todos los actos o sucesos descritos son necesarios para la puesta en práctica del procedimiento). Asimismo, en determinados ejemplos, los actos o sucesos se pueden realizar simultáneamente, por ejemplo, mediante procesamiento de múltiples hilos, procesamiento de interrupciones o múltiples procesadores, en lugar de secuencialmente. Además, si bien determinados aspectos de la presente divulgación se describen como realizados por un único módulo o unidad con fines de claridad, se debería entender que las técnicas de la presente divulgación se pueden llevar a cabo por una combinación de unidades o módulos asociados a un codificador de vídeo.

15 **[0250]** Si bien se han descrito anteriormente combinaciones particulares de diversos aspectos de las técnicas, estas combinaciones se proporcionan meramente para ilustrar ejemplos de las técnicas descritas en esta divulgación. En consecuencia, las técnicas de esta divulgación no deberían limitarse a estas combinaciones ejemplares y pueden abarcar cualquier combinación concebible de los diversos aspectos de las técnicas descritas en esta divulgación.

20 **[0251]** En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, firmware o cualquier combinación de los mismos. Si se implementan en software, las funciones se pueden almacenar en, o transmitir por, un medio legible por ordenador, como una o más instrucciones o código, y ejecutarse mediante una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que correspondan a un medio tangible tal como medios de almacenamiento de datos, o medios de comunicación que incluyan cualquier medio que facilite la transferencia de un programa informático desde un lugar a otro, por ejemplo, de acuerdo a un protocolo de comunicación.

25 **[0252]** De esta manera, los medios legibles por ordenador pueden corresponder en general a (1) medios de almacenamiento tangibles y legibles por ordenador que sean no transitorios o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser medios disponibles cualesquiera a los que se pueda acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en esta divulgación. Un producto de programa informático puede incluir un medio de almacenamiento legible por ordenador y materiales de embalaje.

30 **[0253]** A modo de ejemplo, y no de limitación, tales medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento de disco óptico, almacenamiento de disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que pueda usarse para almacenar código de programa deseado en forma de instrucciones o estructuras de datos y al que pueda accederse mediante un ordenador. Además, cualquier conexión recibe adecuadamente la denominación de medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde una sede de la Red, un servidor u otro origen remoto usando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas se incluyen en la definición de medio.

35 **[0254]** 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 ni 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 el disco compacto (CD), el disco láser, el disco óptico, el disco versátil digital (DVD), el disco flexible y el disco Blu-ray, donde algunos discos reproducen habitualmente los datos magnéticamente, mientras que otros discos reproducen los datos ópticamente con láseres. Las combinaciones de lo anterior deberían incluirse también dentro del alcance de los medios legibles por ordenador.

40 **[0255]** Las instrucciones pueden ser ejecutadas por uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), formaciones lógicas programables in situ (FPGA) u otros circuitos lógicos equivalentes, integrados o discretos. En consecuencia, el término "procesador", como se usa en el presente documento, se puede referir a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento se puede proporcionar dentro de módulos de hardware y/o software dedicados, configurados para la codificación y la decodificación, o incorporados en un códec combinado. Además, las técnicas se podrían implementar totalmente en

uno o más circuitos o elementos lógicos.

5 **[0256]** Las técnicas de esta divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluidos un equipo manual inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). Diversos componentes, módulos o unidades se describen en esta divulgación para enfatizar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no requieren necesariamente su realización mediante diferentes unidades de hardware. En cambio, como se ha descrito anteriormente, diversas unidades se pueden combinar en una unidad de hardware de códec o proporcionarse mediante un grupo de unidades de hardware interoperativas, incluidos uno o más procesadores, como se ha descrito anteriormente, conjuntamente con software y/o firmware adecuados.

10

[0257] Se han descrito diversos aspectos de la divulgación.

REIVINDICACIONES

1. Un procedimiento de decodificación de datos de vídeo divididos en bloques de diferentes tamaños, comprendiendo el procedimiento:

5 construir (800) una lista de patrones de partición que incluye uno o más patrones de partición asociados a un bloque de un primer tamaño, siendo el número de los uno o más patrones de partición directamente proporcional al tamaño del bloque;

10 recibir un índice de la lista de patrones de partición,

determinar (802) un patrón de partición asociado al bloque del primer tamaño, en función del índice recibido de la lista de patrones de partición;

15 determinar (804) un patrón de partición para un bloque de profundidad de un segundo tamaño, basándose en el patrón de partición determinado asociado al bloque del primer tamaño, en donde el segundo tamaño es más grande que el primer tamaño, en donde el bloque del primer tamaño está situado dentro del bloque de profundidad del segundo tamaño y en el que el bloque de profundidad es un bloque de un mapa de profundidad en un marco de codificación de vídeo tridimensional (3D); y

20 decodificar por intrapredicción (806) el bloque de profundidad basándose en el patrón de partición determinado para el bloque de profundidad, sin decodificar por intrapredicción el bloque del primer tamaño usando el patrón de partición determinado, asociado al bloque del primer tamaño, en donde el bloque del primer tamaño se decodifica por intrapredicción como parte de la intrapredicción del bloque de profundidad.

- 25 2. El procedimiento de la reivindicación 1, que comprende además:

30 recibir información que identifica un subbloque dentro del bloque de profundidad, en donde un tamaño del subbloque es el primer tamaño,

en donde determinar el patrón de partición asociado al bloque del primer tamaño comprende determinar el patrón de partición para el subbloque, y

35 en donde determinar el patrón de partición para el bloque de profundidad del segundo tamaño comprende determinar el patrón de partición para el bloque de profundidad del segundo tamaño, basándose en el patrón de partición determinado para el subbloque,

40 en donde la determinación del patrón de partición para el bloque de profundidad del segundo tamaño, basándose en el patrón de partición determinado para el subbloque, comprende preferiblemente extender una línea lineal del patrón de partición para el subbloque a los límites del bloque de profundidad, en donde una línea lineal resultante define el patrón de partición para el bloque de profundidad.

- 45 3. Un procedimiento para codificar datos de vídeo divididos en bloques de diferentes tamaños, comprendiendo el procedimiento:

50 construir (900) una lista de patrones de partición que incluye uno o más patrones de partición asociados a un bloque de un primer tamaño, siendo el número de los uno o más patrones de partición directamente proporcional al tamaño del bloque;

determinar (902) un patrón de partición asociado al bloque del primer tamaño, basándose en la lista de patrones de partición;

55 determinar (904) un patrón de partición para un bloque de profundidad de un segundo tamaño, basándose en el patrón de partición determinado asociado al bloque del primer tamaño, en donde el segundo tamaño es más grande que el primer tamaño, en donde se ubica el bloque del primer tamaño dentro del bloque de profundidad del segundo tamaño, y en el que el bloque de profundidad es un bloque de un mapa de profundidad en un marco de codificación de vídeo tridimensional (3D);

60 codificar por intrapredicción (906) el bloque de profundidad basándose en el patrón de partición determinado para el bloque de profundidad, sin codificar por intrapredicción el bloque del primer tamaño, utilizando el patrón de partición determinado asociado al bloque del primer tamaño, en donde el bloque del primer tamaño se codifica por intrapredicción como parte de la intrapredicción del bloque de profundidad; y

65 señalar un índice de la lista de patrones de partición construida, que identifica el patrón de partición determinado, asociado al bloque del primer tamaño, en donde un decodificador de vídeo usa el índice para decodificar por intrapredicción el bloque de profundidad.

4. El procedimiento de la reivindicación 3, que comprende además:

5 identificar un subbloque dentro del bloque de profundidad, en donde un tamaño del subbloque es el primer tamaño,
 en donde determinar el patrón de partición asociado al bloque del primer tamaño comprende determinar el patrón de partición para el subbloque identificado, y
 10 en donde determinar el patrón de partición para el bloque de profundidad del segundo tamaño comprende determinar el patrón de partición para el bloque de profundidad del segundo tamaño basándose en la partición determinada para el subbloque identificado,
 15 en donde la determinación del patrón de partición para el bloque de profundidad del segundo tamaño basándose en el patrón de partición determinado para el subbloque identificado comprende preferiblemente extender una línea lineal del patrón de partición para el subbloque identificado hasta las fronteras del bloque de profundidad, en donde una línea lineal resultante define el patrón de partición para el bloque de profundidad.

20 5. Un dispositivo para decodificar datos de vídeo divididos en bloques de diferentes tamaños, comprendiendo el dispositivo:

25 una memoria de datos de vídeo que almacena uno o más patrones de partición asociados a un bloque de un primer tamaño; y
 un codificador de vídeo que comprende uno o más procesadores, estando el codificador de vídeo configurado para:
 30 construir una lista de patrones de partición que incluya los uno o más patrones de partición asociados al bloque del primer tamaño, siendo el número de los uno o más patrones de partición directamente proporcional al tamaño del bloque;
 recibir un índice de la lista de patrones de partición,
 35 determinar un patrón de partición asociado al bloque del primer tamaño en función del índice recibido de la lista de patrones de partición;
 40 determinar un patrón de partición para un bloque de profundidad de un segundo tamaño basándose en el patrón de partición determinado, asociado al bloque del primer tamaño, en donde el segundo tamaño es más grande que el primer tamaño, en donde el bloque del primer tamaño está ubicado dentro del bloque de profundidad del segundo tamaño, y en el que el bloque de profundidad es un bloque de un mapa de profundidad en un marco de codificación de vídeo tridimensional (3D); y
 45 decodificar por intrapredicción el bloque de profundidad en función del patrón de partición determinado para el bloque de profundidad sin decodificar por intrapredicción el bloque del primer tamaño utilizando el patrón de partición determinado, asociado al bloque del primer tamaño, en donde el bloque del primer tamaño es decodificado por intrapredicción como parte de la intrapredicción del bloque de profundidad.

50 6. Un dispositivo para codificar datos de vídeo divididos en bloques de diferentes tamaños, comprendiendo el dispositivo:

una memoria de datos de vídeo que almacena uno o más patrones de partición asociados a un bloque de un primer tamaño; y
 55 un codificador de vídeo que comprende uno o más procesadores, estando el codificador de vídeo configurado para:
 60 construir una lista de patrones de partición que incluya los uno o más patrones de partición asociados al bloque del primer tamaño, siendo el número de los uno o más patrones de partición directamente proporcional al tamaño del bloque;
 determinar un patrón de partición asociado al bloque del primer tamaño basándose en la lista de patrones de partición;
 65 determinar un patrón de partición para un bloque de profundidad de un segundo tamaño basándose en el patrón de partición determinado, asociado al bloque del primer tamaño, en donde el segundo tamaño

es más grande que el primer tamaño, en donde el bloque del primer tamaño está ubicado dentro del bloque de profundidad del segundo tamaño, y en el que el bloque de profundidad es un bloque de un mapa de profundidad en un marco de codificación de vídeo tridimensional (3D); y

5 codificar por intrapredicción el bloque de profundidad en función del patrón de partición determinado para el bloque de profundidad sin codificar por intrapredicción el bloque del primer tamaño utilizando el patrón de partición determinado, asociado al bloque del primer tamaño, en donde el bloque del primer tamaño se codifica por intrapredicción como parte de la intrapredicción del bloque de profundidad; y

10 señalar un índice de la lista de patrones de partición construida, que identifica el patrón de partición determinado, asociado al bloque del primer tamaño, en donde un decodificador de vídeo usa el índice para decodificar por intrapredicción el bloque de profundidad.

15 **7.** El dispositivo de la reivindicación 5 o la reivindicación 6, en el que el patrón de partición asociado para un bloque del primer tamaño comprende un patrón de partición asociado a un bloque del tamaño 16x16, en el que el bloque de profundidad del segundo tamaño comprende un bloque de profundidad del tamaño 32x32, y en el que el codificador de vídeo está configurado para determinar el patrón de partición para el bloque de profundidad de tamaño 32x32 en función del patrón de partición determinado, asociado al bloque de tamaño 16x16.

20 **8.** El dispositivo de la reivindicación 5,
en el que el decodificador de vídeo está configurado para recibir información que identifica un subbloque dentro del bloque de profundidad, en donde un tamaño del subbloque es el primer tamaño,

25 en donde, para determinar el patrón de partición asociado al bloque del primer tamaño, el decodificador de vídeo está configurado para determinar el patrón de partición para el subbloque, y

30 en donde, para determinar el patrón de partición para el bloque de profundidad del segundo tamaño, el decodificador de vídeo se configura para determinar el patrón de partición para el bloque de profundidad del segundo tamaño, basándose en el patrón de partición determinado para el subbloque.

9. El dispositivo de la reivindicación 5,
35 en el que, para determinar el patrón de partición para el bloque de profundidad del segundo tamaño, el decodificador de vídeo está configurado para determinar el patrón de partición para el bloque de profundidad del segundo tamaño sin recibir información que indique el patrón de partición para el bloque de profundidad del segundo tamaño.

40 **10.** El dispositivo de la reivindicación 6,
en el que el codificador de vídeo está configurado para identificar un subbloque dentro del bloque de profundidad, en el que un tamaño del subbloque es el primer tamaño,

45 en el que, para determinar el patrón de partición asociado al bloque del primer tamaño, el codificador de vídeo está configurado para determinar el patrón de partición para el subbloque identificado, y

50 en el que, para determinar el patrón de partición para el bloque de profundidad del segundo tamaño, el codificador de vídeo está configurado para determinar el patrón de partición para el bloque de profundidad del segundo tamaño en función de la partición determinada para el subbloque identificado.

11. El dispositivo de la reivindicación 5 o la reivindicación 6, en el que el patrón de partición asociado al bloque del primer tamaño comprende un patrón de partición basado en líneas a partir de una pluralidad de patrones de partición basados en líneas, asociados al bloque del primer tamaño, y en el que el patrón de partición para el bloque de profundidad comprende un patrón de partición basado en líneas para el bloque de profundidad que divide el bloque de profundidad en dos particiones.

55 **12.** El dispositivo de la reivindicación 5 o la reivindicación 6, en el que el dispositivo comprende al menos uno entre: un circuito integrado (IC), un micro-controlador, un dispositivo inalámbrico, un ordenador de escritorio, un ordenador portátil, un ordenador de tableta, un decodificador, un equipo manual de teléfono, un televisor, una cámara, un dispositivo de visualización, un reproductor de medios digitales, una consola de videojuegos o un dispositivo de transmisión continua de vídeo.

65 **13.** Un medio de almacenamiento legible por ordenador que presenta instrucciones almacenadas en el mismo que, al ser ejecutadas por uno o más procesadores de un dispositivo para la codificación de vídeo, hacen que los uno o más procesadores realicen el procedimiento de cualquiera de las reivindicaciones 1 a 4.

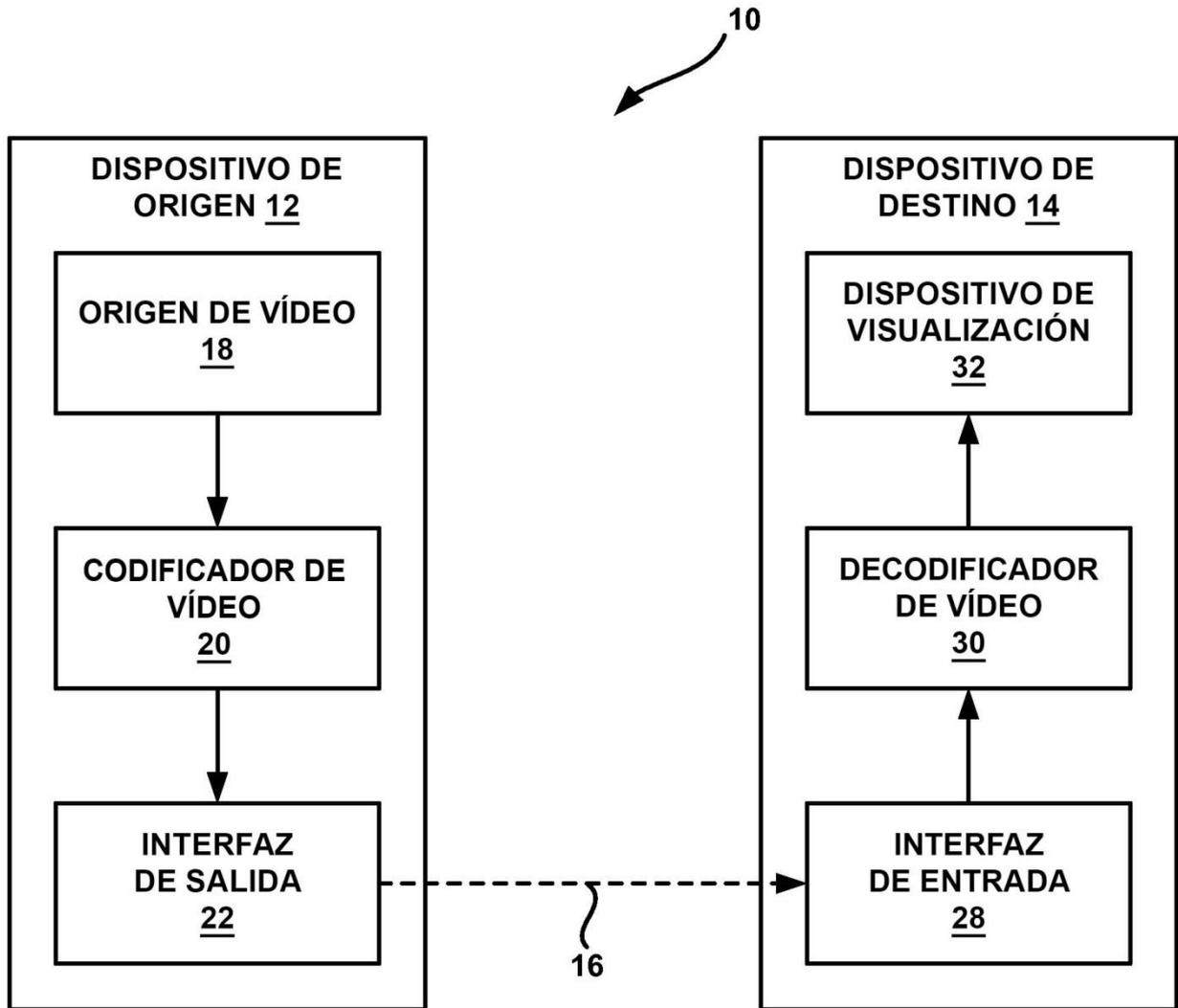


FIG. 1

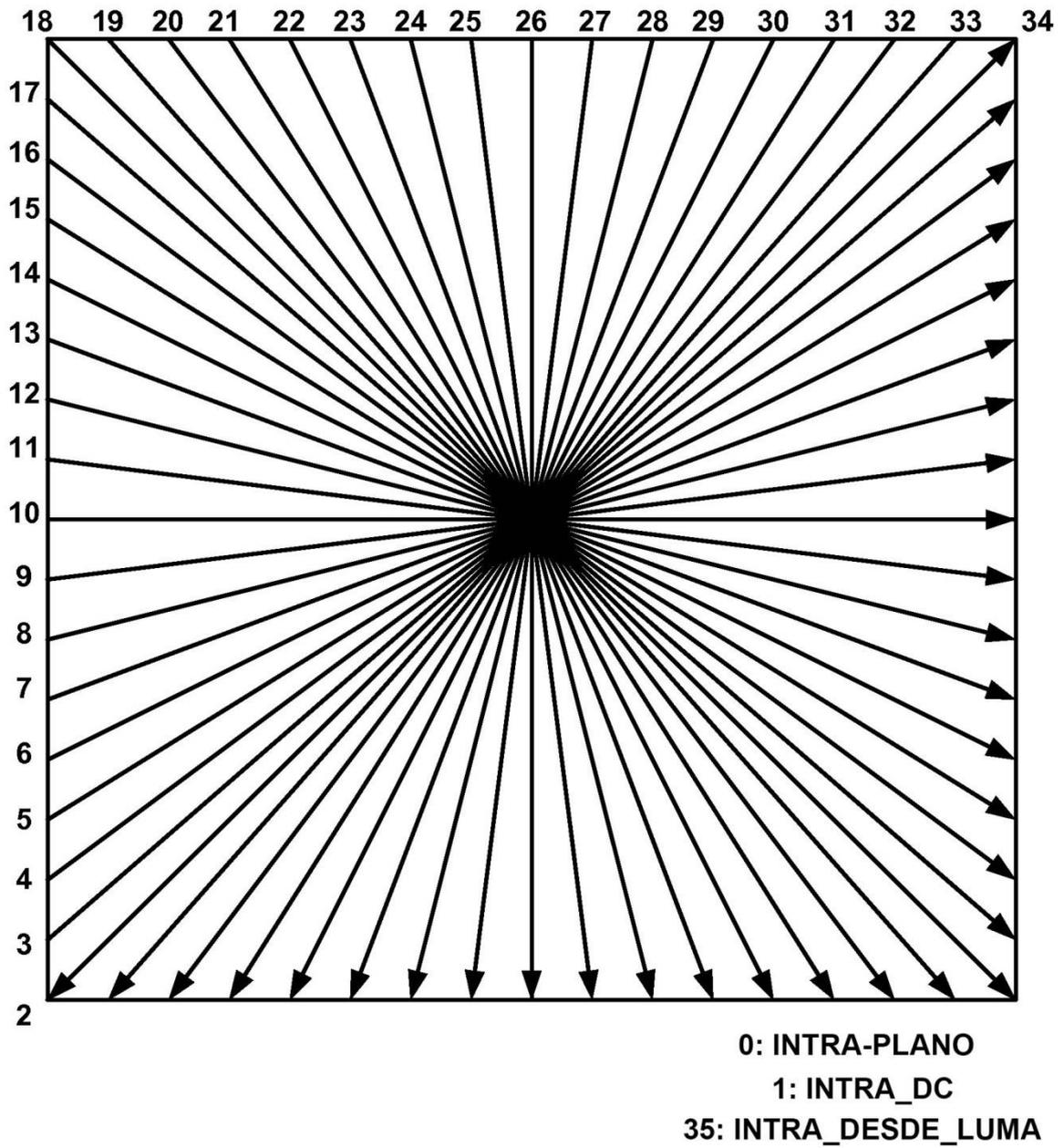


FIG. 2

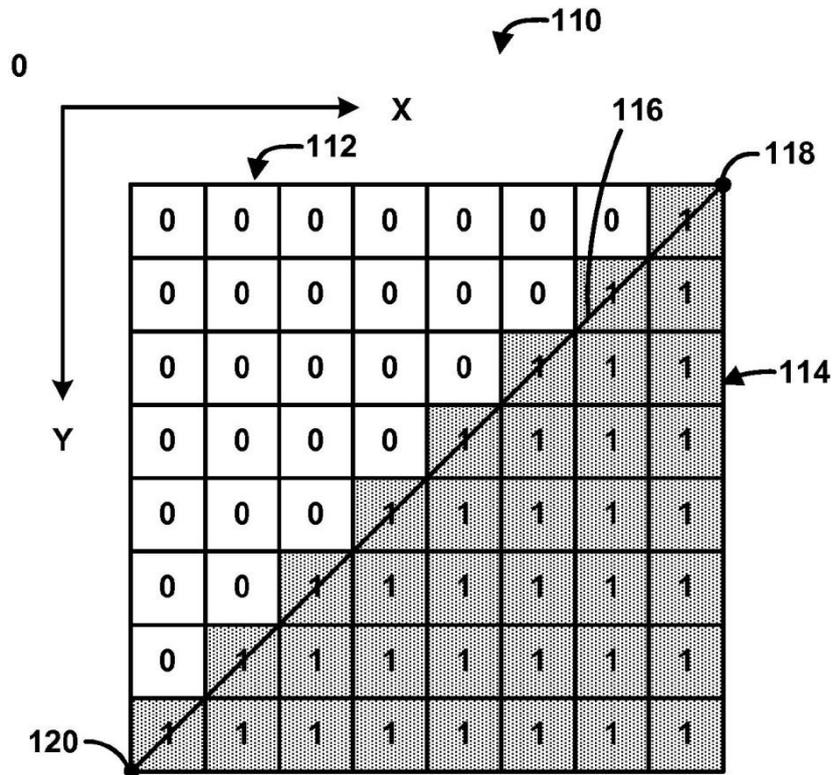


FIG. 3A

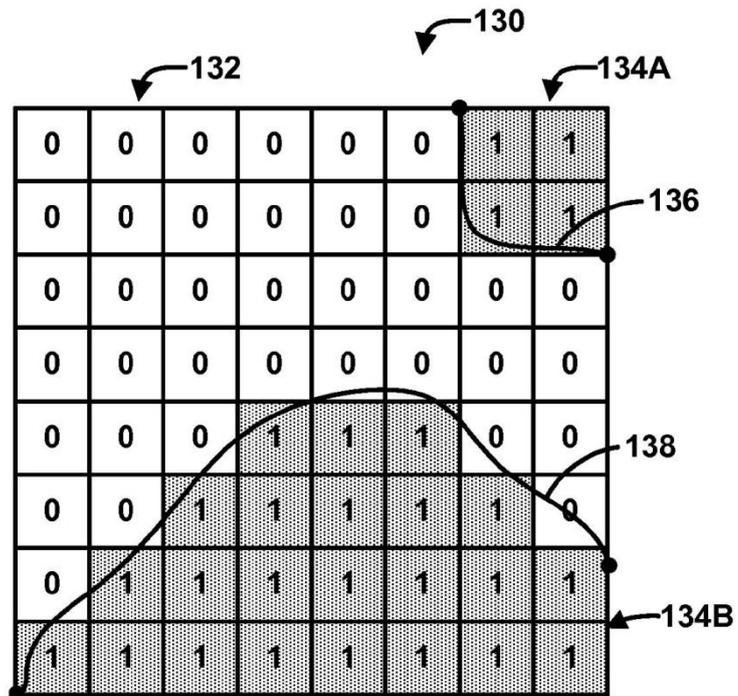
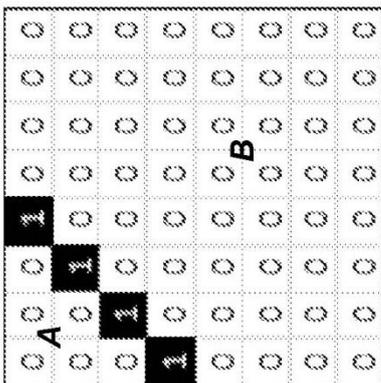
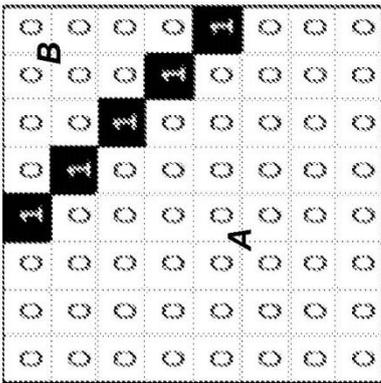


FIG. 3B



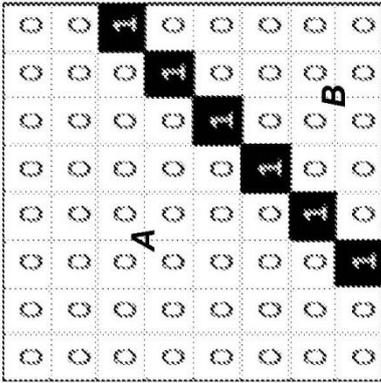
Orientación: 0

FIG. 4A



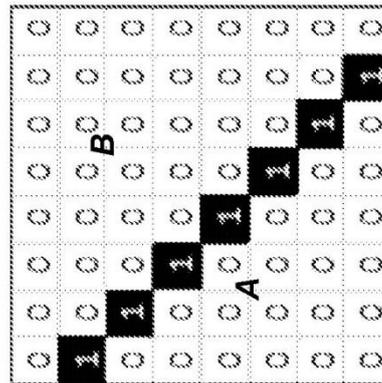
Orientación: 1

FIG. 4B



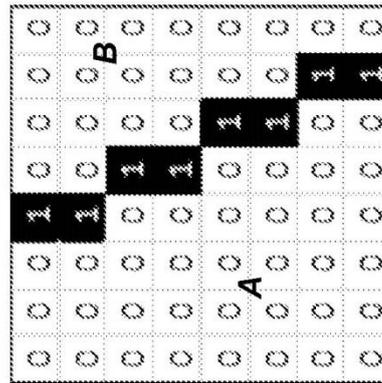
Orientación: 2

FIG. 4C



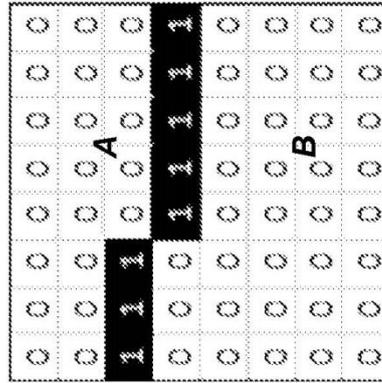
Orientación: 3

FIG. 4D



Orientación: 4

FIG. 4E



Orientación: 5

FIG. 4F

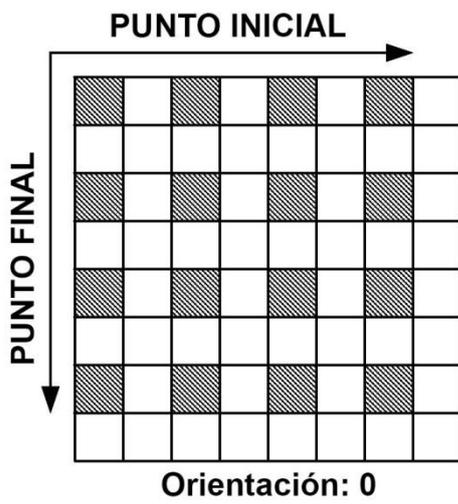


FIG. 5A

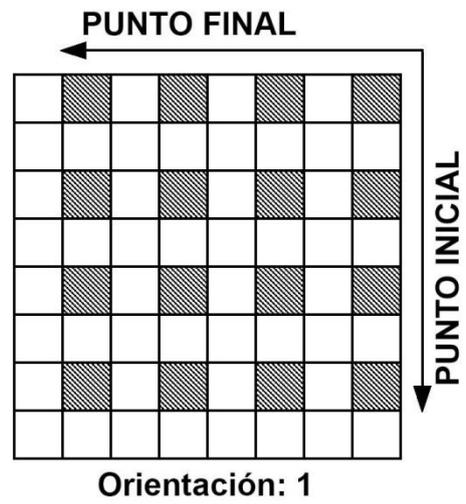


FIG. 5B

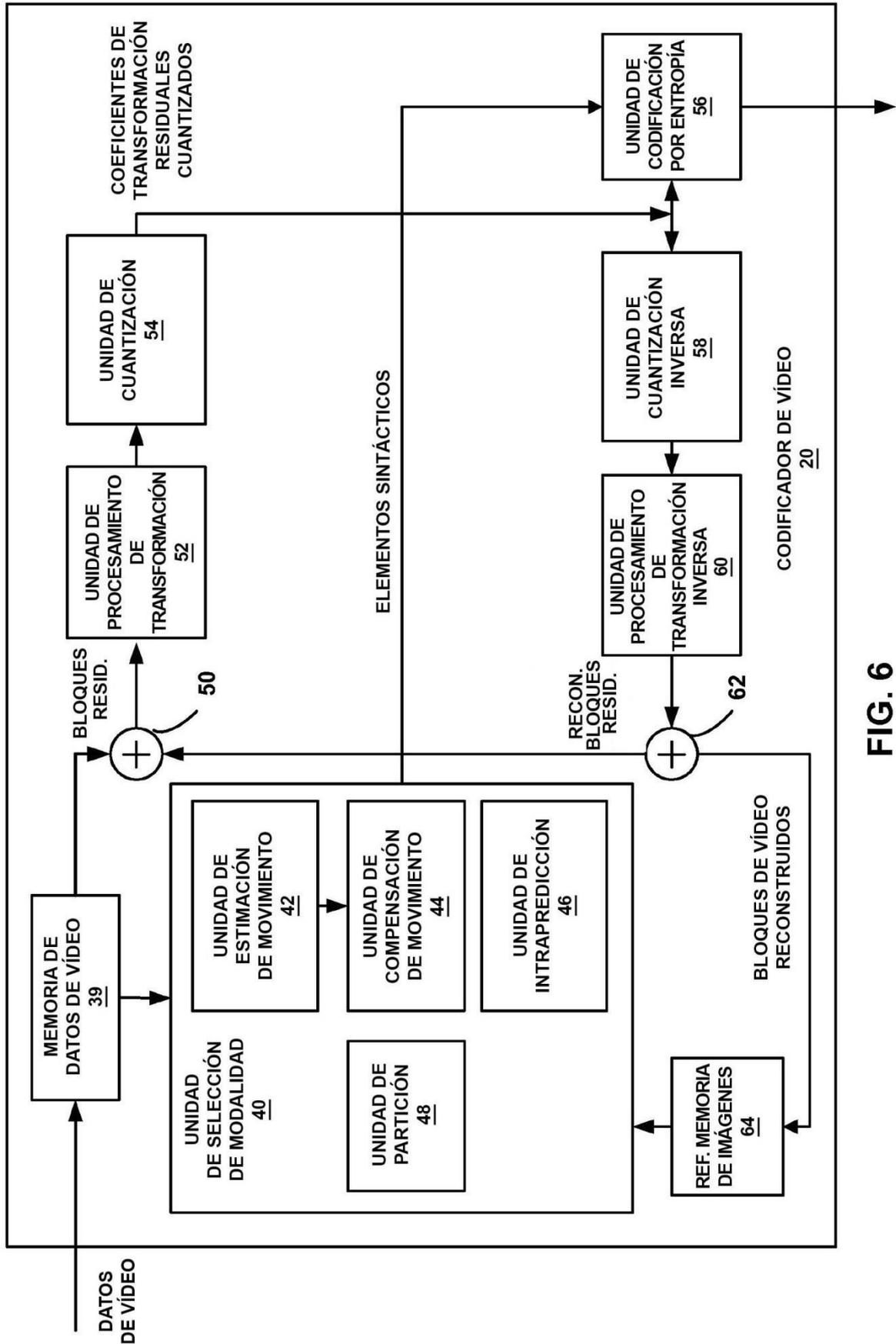


FIG. 6

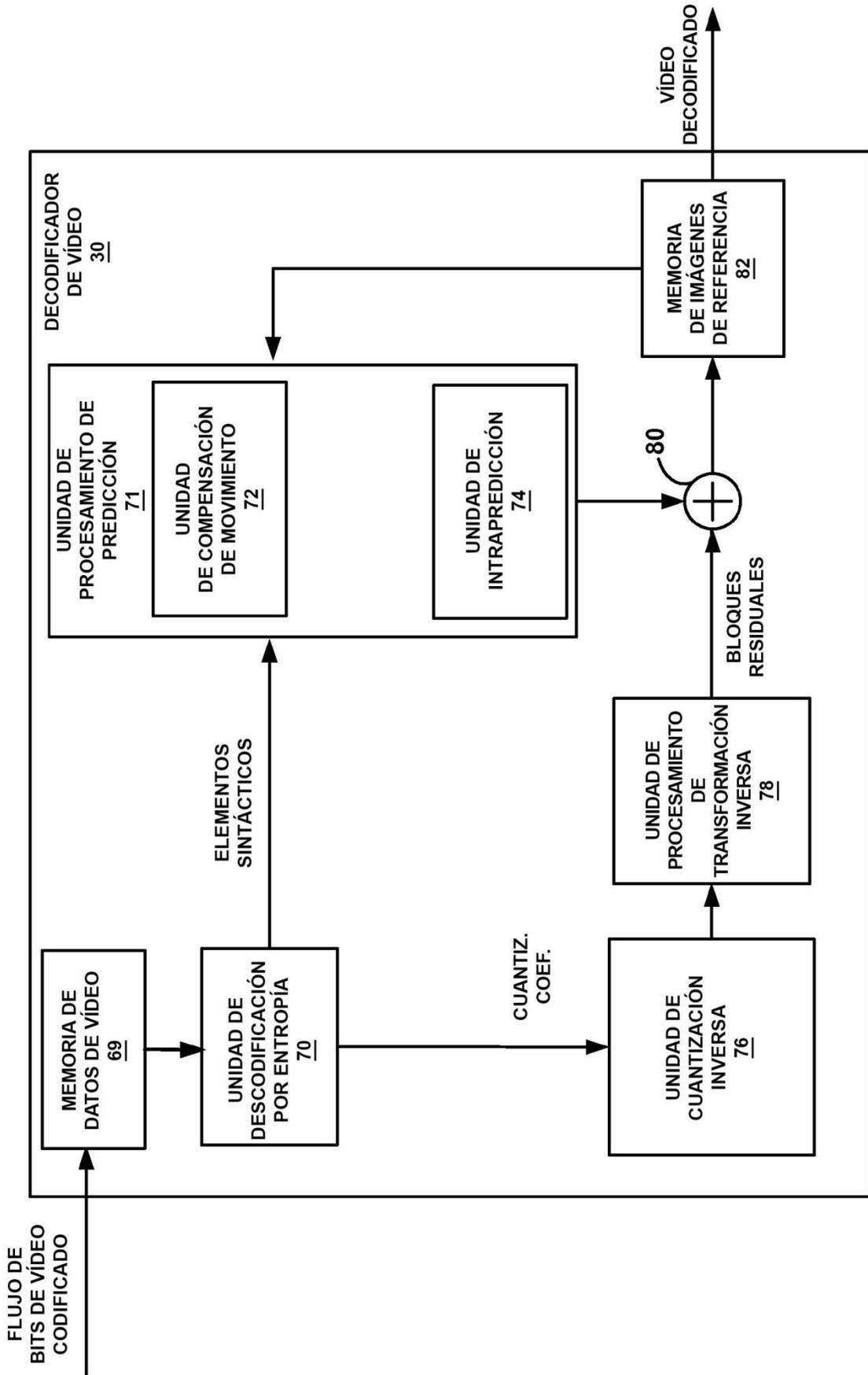


FIG. 7

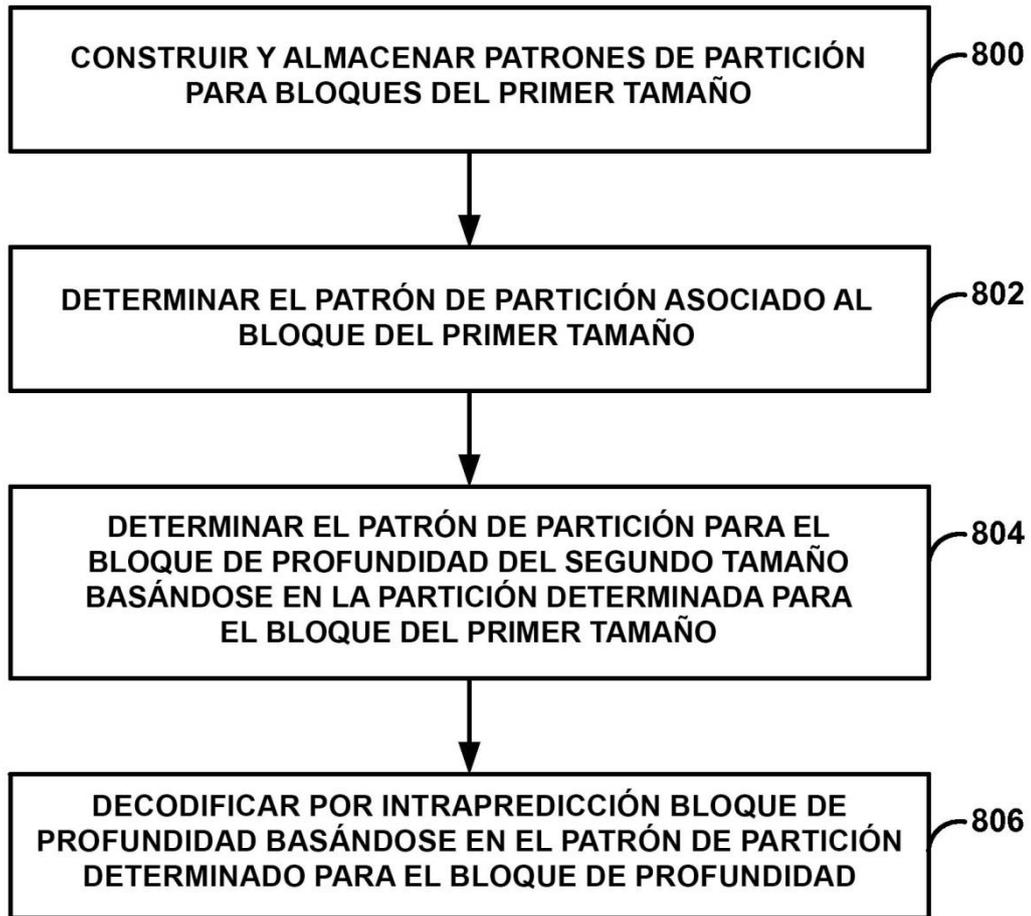


FIG. 8

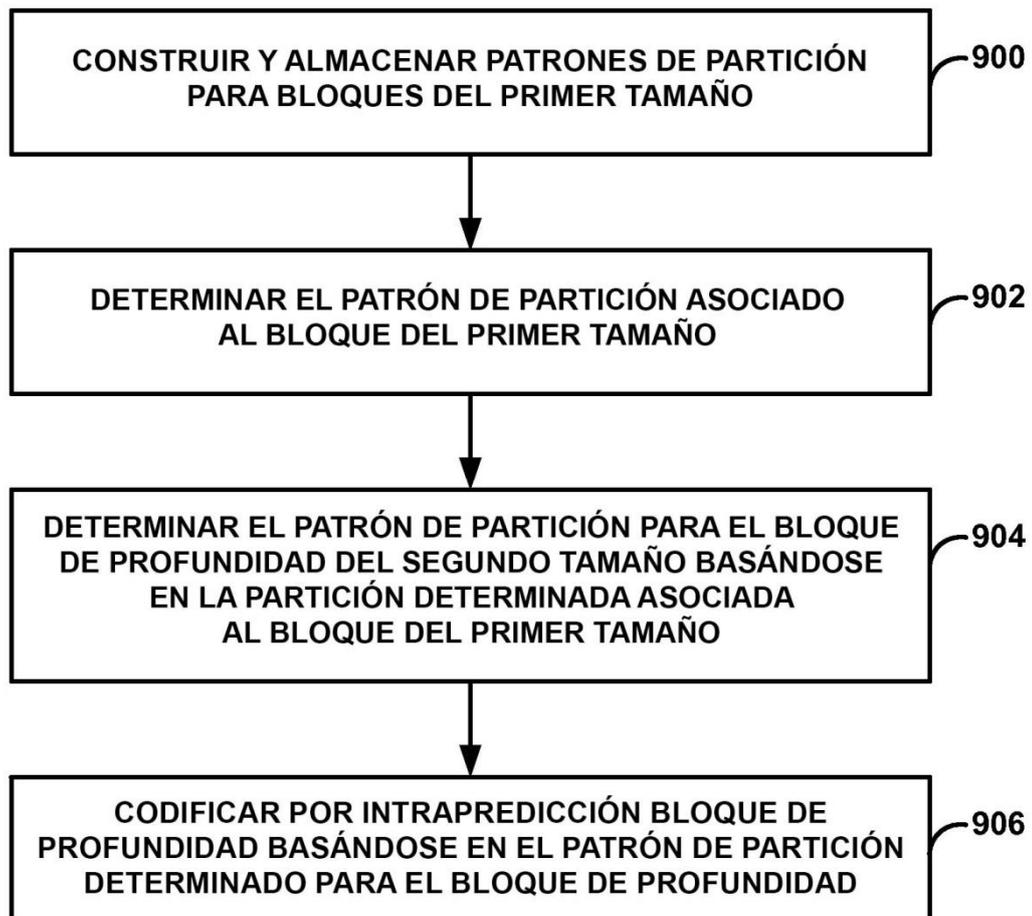


FIG. 9

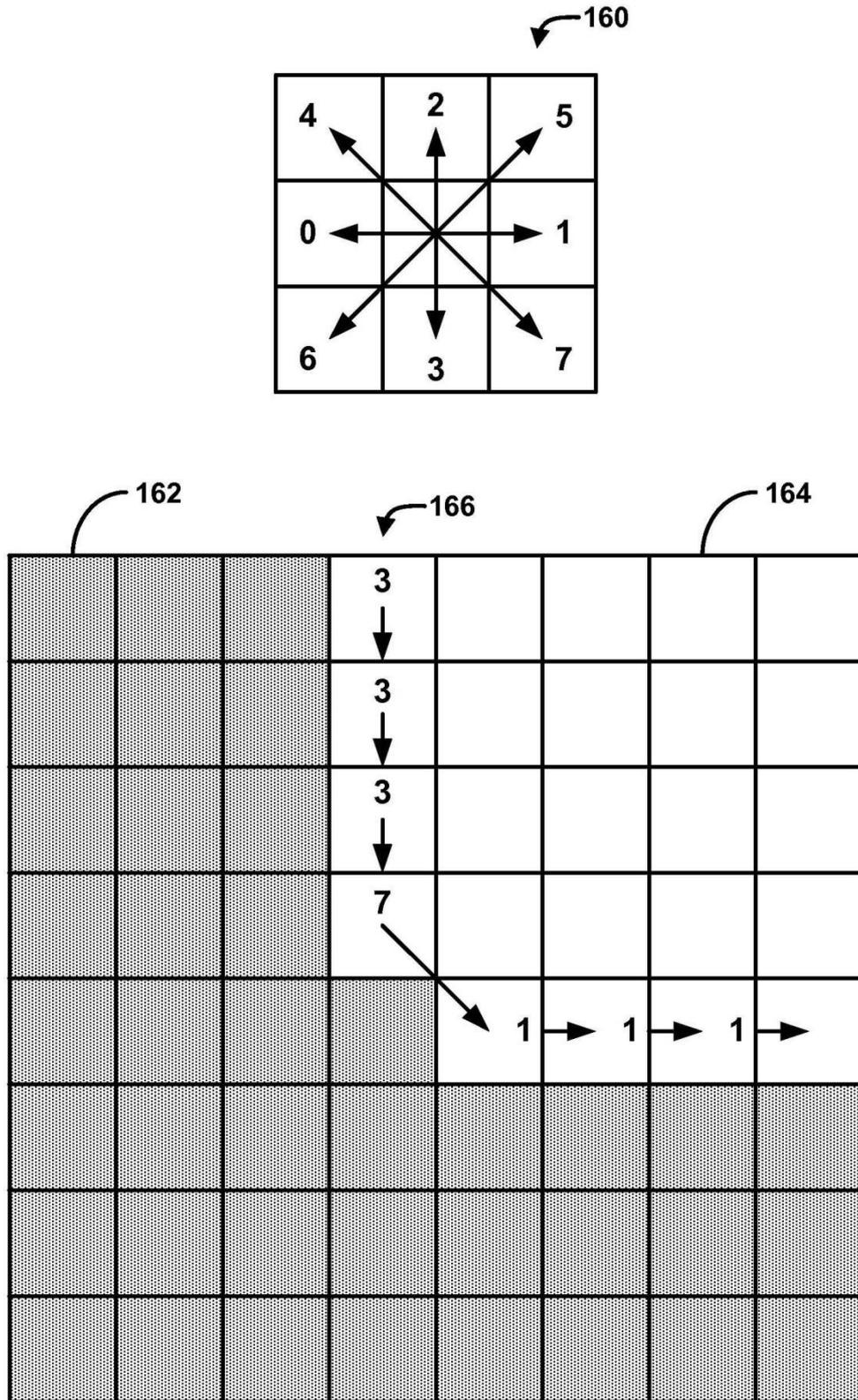


FIG. 10