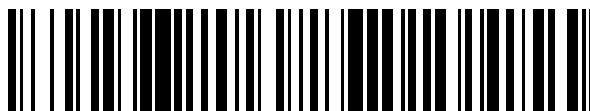


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 673 069**

51 Int. Cl.:

H04N 19/174 (2014.01)

H04N 19/436 (2014.01)

H04N 19/17 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **26.02.2013** **PCT/US2013/027760**

87 Fecha y número de publicación internacional: **17.10.2013** **WO13154687**

96 Fecha de presentación y número de la solicitud europea: **26.02.2013** **E 13709644 (2)**

97 Fecha y número de publicación de la concesión europea: **21.03.2018** **EP 2842312**

54 Título: **Procesamiento paralelo de frente de onda para codificación de vídeo**

30 Prioridad:

11.04.2012 US 201261622974 P
30.04.2012 US 201261640529 P
25.02.2013 US 201313776071

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
19.06.2018

73 Titular/es:

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

72 Inventor/es:

COBAN, MUHAMMED ZEYD;
WANG, YE-KUI y
KARCZEWICZ, MARTA

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 673 069 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

Procesamiento paralelo de frente de onda para codificación de vídeo

5 CAMPO TÉCNICO

[1] Esta divulgación se refiere a la codificación de vídeo.

10 ANTECEDENTES

[2] Las capacidades del vídeo digital pueden incorporarse en una amplia gama de dispositivos, incluyendo televisores digitales, sistemas de radiodifusión directa digital, sistemas de radiodifusión 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 continua de vídeo y similares. Los dispositivos de vídeo digital implementan técnicas de codificación de vídeo, tales como las descritas en las normas definidas por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificación Avanzada de Vídeo (AVC), la norma de Codificación de Vídeo de Alta Eficiencia (HEVC), actualmente en desarrollo, y las ampliaciones de tales normas. Las ampliaciones de normas incluyen, por ejemplo, ampliaciones de la Codificación de Vídeo Escalable (SVC) y la Codificación de Vídeo Multivista (MVC) de H.264/AVC. Los dispositivos de vídeo pueden transmitir, recibir, codificar, descodificar y/o almacenar información de vídeo digital más eficazmente, implementando tales técnicas de codificación de vídeo.

[3] Las técnicas de codificación de vídeo incluyen la predicción espacial (intra-imagen) y/o la predicción temporal (inter-imagen) para reducir o eliminar la redundancia intrínseca en las secuencias de vídeo. Para la codificación de vídeo basada en bloques, un fragmento de vídeo (por ejemplo, una trama de vídeo o una parte de una trama de vídeo) puede dividirse en bloques de vídeo, que también pueden denominarse bloques arbolados, unidades de codificación (CU) y/o nodos de codificación. Los bloques de vídeo en un fragmento intra-codificado (I) de una imagen se codifican mediante predicción espacial con respecto a muestras de referencia en bloques contiguos en la misma imagen. Los bloques de vídeo en un fragmento inter-codificado (P o B) de una imagen pueden usar predicción espacial con respecto a muestras de referencia en bloques contiguos en la misma imagen, o predicción temporal con respecto a muestras de referencia en otras imágenes de referencia. Las imágenes pueden denominarse tramas, y las imágenes de referencia pueden denominarse tramas de referencia.

[4] 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 inter-codificado se codifica de acuerdo con un vector de movimiento que apunta a un bloque de muestras de referencia que forman el bloque predictivo, y los datos residuales que indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intra-codificado se codifica de acuerdo con un modo de intra-codificación y los datos residuales. Para una mayor compresión, los datos residuales pueden transformarse desde el dominio de los píxeles al dominio de las transformaciones, dando como resultado unos coeficientes de transformación residuales, que posteriormente se pueden cuantificar. Los coeficientes de transformación cuantificados, dispuestos inicialmente en una matriz bidimensional, pueden explorarse con el fin de generar un vector unidimensional de coeficientes de transformación, y puede aplicarse la codificación por entropía para lograr aún más compresión.

[5] En JCTVC-H0348 de Lee y otros, "Simplification on tiles and slices" [Simplificación de cuadros y fragmentos], equipo conjunto de colaboración en codificación de vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11 8º congreso: San José, California, EE. UU., del 1 al 10 de febrero de 2012, se divulga un diseño para la simplificación de fragmentos y cuadros, en el que no hay separación entre un cuadro independiente y un cuadro dependiente. Al final de cada cuadro, se termina un fragmento. Por lo tanto, cada fragmento se incluye en un cuadro sin cruzar los límites del cuadro.

55 SUMARIO

[6] En general, esta divulgación describe técnicas para el procesamiento paralelo de frentes de onda de una imagen. En particular, de acuerdo con ciertas técnicas de esta divulgación, un codificador de vídeo puede configurarse para codificar datos de vídeo para una imagen que tiene uno o más frentes de onda, incluyendo cada uno de los frentes de onda uno o más fragmentos completos. Como otro ejemplo, de acuerdo con ciertas técnicas de esta divulgación, un codificador de vídeo puede configurarse para codificar datos de vídeo para una imagen que tiene uno o más fragmentos, incluyendo cada uno de los fragmentos uno o más frentes de onda. En cualquier caso, cada frente de onda en una pluralidad de frentes de onda puede incluir información suficiente para permitir el procesamiento paralelo de los frentes de onda. Por lo tanto, un frente de onda o bien puede comenzar siempre con un encabezado de fragmento, o bien si el frente de onda no comienza con un encabezado de fragmento, entonces el frente de onda pertenece al mismo fragmento que una fila de bloques por encima del frente de onda.

[7] La atención del lector se dirige a las reivindicaciones independientes adjuntas, que definen la invención.

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

BREVE DESCRIPCIÓN DE LOS DIBUJOS

[9]

La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y decodificación de vídeo de ejemplo que puede utilizar técnicas para codificar frentes de onda en paralelo.

La FIG. 2 es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo que puede implementar técnicas para codificar frentes de onda en paralelo.

La FIG. 3 es un diagrama de bloques que ilustra un ejemplo de un decodificador de vídeo que puede implementar técnicas para codificar frentes de onda en paralelo.

La FIG. 4 es un diagrama conceptual que ilustra una imagen de ejemplo dividida en frentes de onda.

La FIG. 5 es un diagrama de flujo que ilustra un proceso de ejemplo mediante el cual un codificador de vídeo puede implementar técnicas de esta divulgación para codificar frentes de onda en paralelo.

La FIG. 6 es un diagrama de flujo que ilustra otro proceso de ejemplo mediante el cual un codificador de vídeo puede implementar técnicas de esta divulgación para codificar frentes de onda en paralelo.

DESCRIPCIÓN DETALLADA

[10] En general, esta divulgación describe técnicas para el procesamiento paralelo de frentes de onda de una imagen. Una imagen puede dividirse en una pluralidad de frentes de onda, donde cada frente de onda puede corresponder a una fila de bloques de la imagen. En ejemplos, los bloques pueden corresponder a unidades de árbol de codificación (CTU) de la imagen, también denominadas unidades de codificación de máximo tamaño (LCU). Un codificador de vídeo, tal como un codificador de vídeo o un decodificador de vídeo, puede codificar los frentes de onda sustancialmente en paralelo. Por ejemplo, el codificador de vídeo puede codificar un bloque de un primer frente de onda de una imagen en paralelo con un bloque de un segundo frente de onda de la imagen. El codificador de vídeo puede inicializar un contexto para un frente de onda actual para realizar una codificación aritmética binaria adaptable al contexto (CABAC) del frente de onda actual basándose en datos de los dos primeros bloques del frente de onda por encima, así como uno o más elementos de un encabezado de fragmento para un fragmento que incluye el primer bloque del frente de onda actual.

[11] Una imagen se puede dividir en múltiples filas de unidades de árbol de codificación (CTU). Cada fila de CTU puede corresponder a un frente de onda respectivo. El procesamiento paralelo de frente de onda ofrece la capacidad de procesar múltiples filas de CTU en paralelo de una manera de frente de onda, donde puede haber un retraso de dos CTU entre el inicio de frentes de onda adyacentes. El codificador de vídeo puede realizar la inicialización de CABAC de un frente de onda (o fila de CTU) siguiente utilizando los estados de contexto después de codificar 2 CTU de una fila de CTU por encima de la fila de CTU siguiente. En otras palabras, antes de comenzar la codificación de un frente de onda actual, un codificador de vídeo puede codificar al menos dos bloques de un frente de onda por encima del frente de onda actual, suponiendo que el frente de onda actual no es la fila de CTU superior de una imagen. Además, el codificador de vídeo puede inicializar un contexto de CABAC para un frente de onda actual después de codificar al menos dos bloques de un frente de onda por encima del frente de onda actual.

[12] Las probabilidades de CABAC se pueden sincronizar con una CTU superior derecha. Debido a que un codificador de vídeo puede procesar frentes de onda en paralelo, el codificador de vídeo puede requerir información del final de la fila de CTU superior para decodificar el comienzo de la segunda fila de CTU. Ejemplos de dicha información pueden incluir información de fragmento, parámetros de cuantificación (QP) y similares. Por ejemplo, si un nuevo fragmento comienza hacia el final de la fila de CTU superior (frente de onda), el codificador de vídeo puede requerir cierta información de la fila de CTU superior antes de codificar la fila de CTU (frente de onda) inmediatamente por debajo. Más específicamente, la información de la fila de CTU superior puede afectar al proceso de decodificación de la fila de CTU por debajo.

[13] En general, las técnicas de esta divulgación están dirigidas a mitigar problemas potenciales causados por fragmentos que comienzan en una posición que está después de la primera CTU de un frente de onda, y continúan en un frente de onda siguiente. En particular, si un fragmento comienza en una posición siguiente a la primera CTU de un frente de onda, e incluye CTU de uno o más frentes de onda siguientes, el codificador de vídeo puede necesitar codificar los respectivos encabezados de fragmento de cada fragmento de un frente de onda actual con el

fin de obtener la información requerida para codificar el frente de onda actual. En un escenario de este tipo, basado en los encabezados de fragmento codificados por un codificador de vídeo, se puede requerir un descodificador de vídeo para examinar cada encabezado de fragmento en una imagen para determinar la información necesaria para descodificar los diversos frentes de onda de la imagen. Ejemplos de dicha información incluyen puntos de entrada de los frentes de onda, parámetros de cuantificación para los frentes de onda, etc. En algunos casos, se puede requerir que el codificador de vídeo asigne los fragmentos según las posiciones en la imagen, tal como asignar el punto inicial y final de cada fragmento de la imagen. Por otro lado, si el codificador de vídeo tiene información para un frente de onda actual desde dentro de 2 CTU a la derecha de la CTU actual de la fila por encima, entonces el codificador de vídeo puede codificar cada frente de onda sin el retardo causado por el desbordamiento de fragmento. Por ejemplo, si un descodificador de vídeo, o un codificador de vídeo configurado para realizar descodificación, tiene acceso a información para un frente de onda actual desde dentro de 2 CTU a la derecha de la CTU actual de la fila por encima, entonces el descodificador de vídeo puede descodificar cada frente de onda sin el retardo causado por el desbordamiento de fragmento.

[14] Para mitigar o evitar retardos de codificación causados por el desbordamiento de fragmento, un codificador de vídeo puede implementar una o más técnicas de esta divulgación para restringir la interacción frente de onda-fragmento de tal manera que si un fragmento comienza en una posición de una fila de CTU distinta del comienzo de la fila de CTU (por ejemplo, el fragmento comienza en el medio de la fila de CTU), entonces el fragmento finaliza dentro de la fila de CTU (por ejemplo, en la última CTU de la fila, o en una CTU que precede a la última CTU de la fila). Por el contrario, el codificador de vídeo puede implementar las técnicas para determinar que un fragmento comienza al comienzo de una fila de CTU (por ejemplo, la primera CTU de la fila forma la primera CTU del fragmento), y que el fragmento incluye todas las CTU de la fila actual y una o más CTU de una o más filas de CTU siguientes. En este escenario, el codificador de vídeo puede permitir el desbordamiento de fragmento, es decir, el codificador de vídeo puede determinar que el fragmento incluye una o más CTU de la una o más filas de CTU siguientes. Una ventaja potencial dada a conocer por las técnicas de esta divulgación es que puede no ser necesario que un descodificador de vídeo dependa de los encabezados de fragmento siguientes cuando se descodifica una imagen en orden de procesamiento paralelo de frente de onda. En cambio, el descodificador puede procesar cada encabezado de fragmento que el descodificador de vídeo encuentra mientras procesa las CTU en orden de procesamiento paralelo de frente de onda.

[15] En algunos ejemplos, el codificador de vídeo puede detectar que un fragmento comienza en el medio o al final de un frente de onda (por ejemplo, en un bloque siguiente al primer bloque del frente de onda), y cruza el límite hasta el frente de onda siguiente. En este caso, el codificador de vídeo puede configurar el fragmento para terminar dentro del (por ejemplo, en el último bloque del) frente de onda en el que comienza el fragmento. Utilizando las configuraciones de fragmento-frente de onda descritas anteriormente, un codificador de vídeo puede garantizar que un frente de onda o bien comienza con un encabezado de fragmento o bien, de manera alternativa, si el frente de onda no comienza con un encabezado de fragmento, entonces el frente de onda pertenece al mismo fragmento que un frente de onda posicionado inmediatamente por encima.

[16] En algunos ejemplos, un codificador de vídeo puede implementar las técnicas para requerir que, si un fragmento comienza al comienzo de un frente de onda y continúa en un frente de onda siguiente, entonces el fragmento debe terminar en el medio de (o de otro modo dentro de) un frente de onda. En combinación con las restricciones descritas anteriormente, el codificador de vídeo puede garantizar que el resto del frente de onda en el que finaliza el primer fragmento incluye uno o más fragmentos completos. Mediante la configuración de fragmentos y frentes de onda de acuerdo con estos requisitos, un codificador de vídeo puede implementar técnicas para realizar el procesamiento paralelo de frente de onda de una imagen de manera más eficiente, tal como mitigando los retardos causados por el desbordamiento de fragmentos que comienzan después de un primer bloque de un frente de onda.

[17] La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y descodificación de vídeo de ejemplo que puede utilizar técnicas para codificar frentes de onda en paralelo. Como se muestra en la FIG. 1, el sistema 10 incluye un dispositivo de origen 12 que proporciona datos de vídeo codificados, a descodificar en un momento posterior mediante 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, incluyendo ordenadores de sobremesa, ordenadores plegables (es decir, portátiles), ordenadores de tableta, descodificadores, equipos telefónicos tales como los denominados teléfonos "inteligentes", los denominados paneles "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, un dispositivo 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.

[18] El dispositivo de destino 14 puede recibir los datos de vídeo codificados a descodificar mediante el medio legible por ordenador 16. El medio legible por ordenador 16 puede comprender cualquier tipo de medio o dispositivo capaz de desplazar 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 que habilita el dispositivo de origen 12 para transmitir datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real. Los datos de vídeo codificados pueden modularse de acuerdo con una norma de comunicación, tal

como un protocolo de comunicación inalámbrica, y transmitirse al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrico o cableado, tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión físicas. 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 al dispositivo de destino 14.

[19] 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 del dispositivo de almacenamiento mediante una interfaz de entrada. El dispositivo de almacenamiento puede incluir cualquiera entre una diversidad de medios de almacenamiento de datos, de acceso distribuido o local, tales como una unidad de disco duro, discos Blu-ray, discos DVD, discos CD-ROM, memoria flash, memoria volátil o no volátil u otros medios adecuados cualesquiera de almacenamiento digital para almacenar datos de vídeo codificados. En un ejemplo adicional, el dispositivo de almacenamiento puede corresponder a un servidor de archivos o a otro dispositivo de almacenamiento intermedio que pueda almacenar los datos de vídeo codificados generados por el dispositivo de origen 12. El dispositivo de destino 14 puede acceder a datos de vídeo almacenados del dispositivo de almacenamiento mediante transmisión continua o descarga. El servidor de archivos 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. Ejemplos de servidores de archivos incluyen un servidor web (por ejemplo, para un sitio web), un servidor FTP, dispositivos de almacenamiento conectado en red (NAS) o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificados mediante cualquier conexión de datos estándar, incluyendo una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión Wi-Fi), una conexión cableada (por ejemplo, DSL, módem de cable, etc.), o una combinación de ambos que sea adecuada para acceder a datos de vídeo codificados almacenados en un servidor de archivos. 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.

[20] Las técnicas de esta divulgación no están limitadas necesariamente a aplicaciones o configuraciones inalámbricas. Las técnicas pueden aplicarse a la codificación de vídeo como apoyo a cualquiera de una diversidad de aplicaciones multimedia, 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 el flujo adaptativo dinámico sobre HTTP (DASH), el vídeo digital que se codifica en un medio de almacenamiento de datos, la decodificación de vídeo digital almacenado en un medio de almacenamiento de datos, u otras aplicaciones. En algunos ejemplos, el sistema 10 puede configurarse para admitir la transmisión de vídeo unidireccional o bidireccional para admitir aplicaciones tales como la transmisión continua de vídeo, la reproducción de vídeo, la difusión de vídeo y/o la videotelefonía.

[21] En el ejemplo de la FIG. 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 con esta divulgación, el codificador de vídeo 20 del dispositivo de origen 12 puede estar configurado para aplicar las técnicas para codificar frentes de onda en paralelo. 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 una fuente de vídeo externa 18, tal como una cámara externa. Asimismo, el dispositivo de destino 14 puede interactuar con un dispositivo de visualización externo, en lugar de incluir un dispositivo de visualización integrado.

[22] El sistema 10 ilustrado de la FIG. 1 es simplemente un ejemplo. Las técnicas para codificar frentes de onda en paralelo pueden realizarse mediante cualquier dispositivo de codificación y/o decodificación de vídeo digital. Aunque, en general, las técnicas de esta divulgación se llevan a cabo mediante un dispositivo de codificación de vídeo, las técnicas también pueden llevarse a cabo mediante un codificador/descodificador de vídeo, denominado habitualmente "CODEC". Además, las técnicas de esta divulgación también pueden llevarse a cabo mediante un pre-procesador 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 de decodificación de vídeo. Por lo tanto, el sistema 10 puede admitir una transmisión de vídeo unidireccional o bidireccional entre los dispositivos de vídeo 12, 14, por ejemplo, para la transmisión continua de vídeo, la reproducción de vídeo, la difusión de vídeo o la videotelefonía.

[23] 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 grabado previamente y/o una interfaz de alimentación de vídeo para recibir vídeo de un proveedor de contenidos de vídeo. De forma alternativa adicional, el origen de vídeo 18 puede generar datos, basados en gráficos por 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 denominados 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 pueden aplicarse a aplicaciones inalámbricas y/o cableadas. En cada caso, el vídeo capturado, pre-capturado o generado por ordenador puede codificarse mediante el codificador de vídeo 20. La información de vídeo codificada puede entonces emitirse mediante la interfaz de salida 22 hacia un medio legible por ordenador 16.

[24] El medio legible por ordenador 16 puede incluir medios transitorios, tales como una emisión inalámbrica o una transmisión de red cableada, o medios de almacenamiento (es decir, medios de almacenamiento no transitorio), 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 manera similar, un dispositivo informático de una instalación de producción de un medio, tal como una instalación 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, puede entenderse que el medio legible por ordenador 16 incluye uno o más medios legibles por ordenador de varias formas, en varios ejemplos.

[25] La interfaz de entrada 28 del dispositivo de destino 14 recibe información desde el medio legible por ordenador 16. La información del medio legible por ordenador 16 puede incluir información sintáctica definida por el codificador de vídeo 20, que también se usa por el descodificador de vídeo 30, que incluye elementos sintácticos que describen características y/o procesamiento de bloques y otras unidades codificadas, por ejemplo, grupos de imágenes (GOP). El dispositivo de visualización 32 muestra los datos de vídeo descodificados a un usuario, y puede comprender cualquiera de diversos 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.

[26] El codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con una norma de codificación de vídeo, tal como la norma de codificación de vídeo de alta eficiencia (HEVC), actualmente en fase de desarrollo, y pueden ajustarse al modelo de prueba de la HEVC (HM). De forma alternativa, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo a otras normas privadas o industriales, tales como la norma ITU-T H.264, alternativamente denominada MPEG-4, Parte 10, Codificación Avanzada de Vídeo (AVC), o ampliaciones de dichas normas. Sin embargo, las técnicas de esta divulgación no están limitadas a ninguna norma de codificación particular. Otros ejemplos de normas de codificación de vídeo incluyen MPEG-2 e ITU-T H.263. Aunque no se muestra en la FIG. 1, en algunos aspectos, cada uno del codificador de vídeo 20 y el descodificador de vídeo 30 pueden estar integrados con un codificador y un descodificador de audio, y pueden incluir unidades MUX-DEMUX apropiadas, 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 ajustarse al protocolo de multiplexación ITU H.223 o a otros protocolos, tales como el protocolo de datagramas de usuario (UDP).

[27] La norma ITU-T H.264/MPEG-4 (AVC) se formuló por el Grupo de Expertos en Codificación de Vídeo de ITU-T (VCEG) junto con el Grupo de Expertos en Imágenes en Movimiento de ISO/IEC (MPEG), como el producto de una asociación colectiva conocida como Equipo Mixto de Vídeo (JVT, por sus siglas en inglés). En algunos aspectos, las técnicas descritas en esta divulgación pueden aplicarse a dispositivos que se ajustan en general a la norma H.264. La norma H.264 se describe en la Recomendación de la ITU-T H.264, Codificación Avanzada de Vídeo para servicios audiovisuales genéricos, por el Grupo de Estudio de la ITU-T, y con fecha de marzo de 2005, que puede denominarse en el presente documento norma H.264 o memoria descriptiva H.264, o norma o memoria descriptiva H.264/AVC. El Equipo Mixto de Vídeo (JVT) continúa trabajando en ampliaciones de H.264/MPEG-4 AVC.

[28] El codificador de vídeo 20 y el descodificador de vídeo 30 pueden implementarse como cualquiera entre una variedad de circuitos de codificadores adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), matrices de puertas programables por capo (FPGA), lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. Cuando las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio adecuado no transitorio, legible por ordenador, y ejecutar las instrucciones en hardware mediante uno o más procesadores para realizar las técnicas de esta divulgación. Cada uno del codificador de vídeo 20 y el descodificador de vídeo 30 pueden estar incluidos en uno o más codificadores o descodificadores, cualquiera de los cuales puede estar integrado como parte de un codificador/descodificador (CODEC) combinado en un dispositivo respectivo.

[29] El equipo JCT-VC está trabajando en el desarrollo de la norma HEVC. Los esfuerzos de normalización de HEVC se basan en un modelo en evolución de un dispositivo de codificación de vídeo denominado modelo de prueba de HEVC (HM). El HM supone varias capacidades adicionales de los dispositivos de codificación de vídeo respecto a dispositivos existentes de acuerdo, por ejemplo, a la norma ITU-T H.264/AVC. Por ejemplo, mientras que la norma H.264 proporciona nueve modos de codificación de intra-predicción, el HM puede proporcionar hasta treinta y tres modos de codificación de intra-predicción.

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

[31] 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 o no en varias sub-CU. Los elementos sintácticos para una CU pueden definirse de manera recursiva y pueden depender de si la CU está dividida en varias sub-CU. Si una CU no está dividida adicionalmente, se denomina CU hoja. En esta divulgación, cuatro sub-CU de una CU hoja también se denominarán CU hojas incluso aunque no haya una división explícita de la CU hoja original. Por ejemplo, si una CU con un tamaño de 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.

[32] Una CU tiene un propósito similar a un macrobloque de la norma H.264, excepto que una CU no tiene una distinción de tamaño. Por ejemplo, un bloque arbolado puede dividirse en cuatro nodos secundarios (también denominados sub-CU) y cada nodo secundario puede a su vez ser un nodo principal y dividirse en otros cuatro nodos secundarios. Un nodo secundario final, no dividido, denominado un 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 que puede dividirse un bloque arbolado, denominado profundidad de CU máxima, y también pueden definir un tamaño mínimo de los nodos de codificación. Por consiguiente, un flujo de bits también puede definir una unidad de codificación de mínimo tamaño (SCU). Esta divulgación utiliza el término "bloque" para referirse a cualquiera de 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).

[33] 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. Los modos de división pueden diferir en función de si la CU está codificada en modo de salto o directo, codificada en modo de intra-predicción o codificada en modo de inter-predicción. 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).

[34] La norma HEVC admite transformaciones de acuerdo con las TU, que pueden ser diferentes para diferentes CU. El tamaño de las TU típicamente se basa en el tamaño de las PU de una CU dada, definida para una CTU dividida, aunque puede que no siempre sea 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 mediante una estructura de árbol cuádruple conocida como "árbol cuádruple residual" (RQT). Los nodos hoja 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 cuantificarse.

[35] 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 en intra-modo, pueden incluirse datos para la PU en un árbol cuádruple residual (RQT), que pueden incluir datos que describen un modo de intra-predicción para una TU correspondiente a la PU. Como otro ejemplo, cuando la PU está codificada en inter-modo, 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, una componente horizontal del vector de movimiento, una 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, la Lista 0, la Lista 1 o la Lista C) para el vector de movimiento.

[36] 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 analizado 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 más sub-TU. Cuando una TU no se divide adicionalmente, puede denominarse una TU hoja. En general, en lo que respecta a la intra-codificación, todas las TU hoja que pertenecen a una CU hoja comparten el mismo modo de intra-predicción. Es decir, el mismo modo de intra-predicción se aplica en general para calcular valores predichos para todas las TU de una CU hoja. En lo que respecta a la intra-codificación, un codificador de vídeo puede calcular un valor residual para cada TU hoja usando el modo de intra-predicción, 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 intra-codificación, una PU puede estar coubicada 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.

[37] Además, las TU de las CU hojas también pueden asociarse a respectivas estructuras de datos de árbol cuádruple, denominadas árboles cuádruales residuales (RQT). Es decir, una CU hoja puede incluir un árbol cuádruple que indica cómo la CU hoja está dividida en varias 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 CTU). 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.

[38] Una secuencia de vídeo incluye típicamente una serie de tramas o imágenes de vídeo. 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 un encabezado del GOP, en un encabezado de una o más de las imágenes o en otras ubicaciones, que describen un cierto número de imágenes incluidas en el GOP. Cada fragmento de una imagen puede incluir datos sintácticos de fragmento que describen un modo de codificación para el fragmento respectivo. Un codificador de vídeo 20 actúa típicamente 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 presentar tamaños fijos o variables y pueden diferir en tamaño de acuerdo con una norma de codificación especificada.

[39] En un ejemplo, el HM admite la predicción en diversos tamaños de PU. Suponiendo que el tamaño de una CU particular es $2N \times 2N$, el HM admite intra-predicción en tamaños de PU de $2N \times 2N$ o $N \times N$ e inter-predicción en tamaños de PU simétricos de $2N \times 2N$, $2N \times N$, $N \times 2N$ o $N \times N$. El HM también admite la división asimétrica para la inter-predicción 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 en el 25% y el 75%. La parte de la CU correspondiente a la división del 25 % está indicada por una "n" seguida de una indicación de "arriba", "abajo", "izquierda" o "derecha". Así pues, por ejemplo, " $2N \times nU$ " 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.

[40] En esta divulgación, " $N \times N$ " 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 tamaño 16×16 tendrá 16 píxeles en la dirección vertical ($y = 16$) y 16 píxeles en la dirección horizontal ($x = 16$). Asimismo, un bloque de tamaño $N \times N$ presenta, en general, N píxeles en una dirección vertical y N píxeles en una dirección horizontal, donde N representa un valor entero no negativo. Los píxeles en un bloque pueden disponerse en filas y columnas. Además, no es necesario que los bloques presenten necesariamente el mismo número de píxeles en la dirección horizontal y en la dirección vertical. Por ejemplo, los bloques pueden comprender $N \times M$ píxeles, donde M no es necesariamente igual a N .

[41] Tras la codificación de intra-predicción o inter-predicción mediante 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 modo 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, a los datos de vídeo residuales. Los datos residuales pueden corresponder a diferencias de píxeles entre píxeles de la imagen no codificada y los 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.

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

[43] Después de la cuantificación, el codificador de vídeo puede explorar los coeficientes de transformación, produciendo un vector unidimensional a partir de la matriz bidimensional que incluye los coeficientes de transformación cuantificados. La exploración puede estar diseñada para colocar los coeficientes de energía más alta (y por lo tanto de menor frecuencia) en la parte frontal de la matriz y para colocar los coeficientes de energía más baja (y por lo tanto de mayor frecuencia) en la parte posterior de la matriz. En algunos ejemplos, el codificador de vídeo 20 puede usar un orden de exploración predefinido para explorar los coeficientes de transformación cuantificados, para producir un vector en serie que pueda ser codificado por entropía. En otros ejemplos, el codificador de vídeo 20 puede realizar una exploración adaptativa. Después de explorar los coeficientes de transformación cuantificados para formar un vector unidimensional, el codificador de vídeo 20 puede realizar la codificación por entropía del vector unidimensional, por ejemplo, de acuerdo con la codificación de longitud variable adaptable al contexto (CAVLC), la codificación aritmética binaria adaptable al contexto (CABAC), la codificación aritmética binaria adaptable al contexto basada en la sintaxis (SBAC), la codificación por entropía por división de intervalos de probabilidad (PIPE) u otros procedimientos 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.

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

[45] El codificador de vídeo 20 y el decodificador de vídeo 30 pueden utilizar procesamiento paralelo de frente de onda (WPP) para codificar y decodificar imágenes, respectivamente. Para codificar una imagen utilizando WPP, un codificador de vídeo, tal como el codificador de vídeo 20 y el decodificador de vídeo 30, puede dividir las unidades de árbol de codificación (CTU) de la imagen en una pluralidad de frentes de onda. Cada frente de onda puede corresponder a una fila de CTU diferente en la imagen. El codificador de vídeo puede comenzar a codificar un frente de onda superior, por ejemplo, utilizando un primer núcleo o hilo del codificador. Después de que el codificador de vídeo haya codificado dos o más CTU del frente de onda superior, el codificador de vídeo puede comenzar a codificar un frente de onda segundo hasta el superior en paralelo con la codificación del frente de onda superior, por ejemplo, utilizando un segundo núcleo o hilo paralelo del codificador. Después de que el codificador de vídeo haya codificado dos o más CTU del frente de onda segundo hasta el superior, el codificador de vídeo puede comenzar a codificar un frente de onda tercero hasta el superior en paralelo con la codificación de los frentes de onda superiores, por ejemplo, utilizando un tercer núcleo o hilo paralelo del codificador. Este patrón puede continuar hacia abajo por los frentes de onda de la imagen.

[46] Esta divulgación se refiere a un conjunto de CTU que un codificador de vídeo está codificando simultáneamente, usando WPP, como un "grupo de CTU". Así pues, cuando el codificador de vídeo está utilizando WPP para codificar una imagen, cada una de las CTU del grupo de CTU puede estar en un frente de onda diferente de la imagen y cada una de las CTU del grupo de CTU puede estar desplazada verticalmente de una CTU en un frente de onda por encima respectivo por al menos dos columnas de CTU de la imagen.

[47] Además, cuando se codifica la imagen utilizando WPP, el codificador de vídeo puede utilizar información asociada con una o más CU contiguas espacialmente fuera de una CTU particular para realizar una intra-predicción o una inter-predicción en una CU particular en la CTU particular, siempre que las CU contiguas espacialmente estén a la izquierda, por encima a la izquierda, por encima, o por encima a la derecha de la CTU particular. Cuando la una o más CU contiguas espacialmente están por encima a la derecha de la CTU particular, también se supone que la una o más CU contiguas espacialmente se codificaron previamente. Si la CTU particular es la CTU más a la izquierda en un frente de onda distinto del frente de onda superior, el codificador de vídeo puede utilizar información asociada con la primera y/o segunda CTU de un frente de onda adyacente (por ejemplo, el frente de onda posicionado inmediatamente por encima) para seleccionar un contexto de codificación para la codificación por entropía de los elementos sintácticos de la CTU particular. Si la CTU particular no es la CTU más a la izquierda en el frente de onda, el codificador de vídeo puede seleccionar información asociada con una CU contigua espacialmente, que está posicionada a la izquierda, por encima a la izquierda, por encima y/o por encima a la derecha de la CTU particular para seleccionar un contexto de codificación para la codificación por entropía de un elemento sintáctico de la CTU particular. De esta forma, el codificador de vídeo puede inicializar estados de codificación por entropía (por ejemplo, CABAC) de un frente de onda basándose en los estados de codificación por entropía del frente de onda posicionado inmediatamente por encima después de codificar dos o más CTU del frente de onda posicionado inmediatamente por encima.

[48] De manera adicional, un codificador de vídeo puede dividir una imagen en fragmentos. En general, cada fragmento se codifica por entropía de manera individual, de tal manera que los contextos se restablecen al comienzo de la codificación de un nuevo fragmento. El codificador de vídeo 20, o una unidad de post-procesamiento del dispositivo de origen 12 (tal como una unidad de encapsulación, no mostrada en la FIG. 1), puede encapsular fragmentos en unidades de capa de abstracción de red (NAL) respectivas. Por ejemplo, una unidad de NAL puede incluir un encabezado de NAL y una carga útil que representa uno o más fragmentos codificados. Para delimitar fragmentos codificados entre sí, el codificador de vídeo 20 puede incluir encabezados de fragmento dentro de la carga útil de la unidad de NAL para indicar el comienzo de un fragmento. De manera adicional, el codificador de vídeo 20 puede incluir uno o más símbolos de fin de fragmento dentro de la carga útil de la unidad de NAL para indicar el final de fragmentos codificados diferentes.

[49] El codificador de vídeo 20 puede dividir una imagen dada en fragmentos de diferentes longitudes. En otras palabras, diferentes fragmentos de una imagen particular pueden incluir o de otro modo corresponder a números variables de CTU. En consecuencia, el codificador de vídeo 20 puede generar diferentes unidades de NAL para incluir diferentes números de fragmentos codificados.

[50] De manera correspondiente, el descodificador de vídeo 30 puede descodificar por entropía la imagen fragmento a fragmento. Más específicamente, el dispositivo de origen 22 puede utilizar la interfaz de salida 22 para transmitir las unidades de NAL a la interfaz de entrada 28 del dispositivo de destino 14. De manera alternativa, la interfaz de salida 22 puede generar unidades de NAL en un medio legible por ordenador, tal como un disco o memoria legible por ordenador, por ejemplo, memoria magnética o memoria flash. El descodificador de vídeo 30 puede recibir las unidades de NAL a través de la interfaz de entrada 28, y extraer cada fragmento codificado utilizando la información de división de fragmento (por ejemplo, encabezados de fragmento y/o símbolos de fin de fragmento) incluida. A su vez, el descodificador de vídeo 30 puede descodificar por entropía cada fragmento extraído, y reconstruir la imagen fragmento a fragmento.

[51] En el contexto de WPP, bajo algunas circunstancias, un codificador de vídeo podría no ser capaz de seleccionar contextos de codificación a lo largo de los límites de fragmento. Por ejemplo, si la información de contexto para una CTU particular pertenece a un fragmento diferente de la CTU posicionada por encima a la derecha de la CTU particular, el codificador de vídeo puede no tener acceso a la información necesaria para codificar la CTU particular. Más específicamente, en términos de posicionamiento dentro de frentes de onda, el encabezado de fragmento para la CTU particular puede no codificarse cuando el codificador de vídeo alcanza la CTU particular. Por ejemplo, el encabezado de fragmento puede estar posicionado en un frente de onda que está inmediatamente por encima del frente de onda de la CTU, y el encabezado de fragmento puede estar posicionado más de dos bloques a la derecha en comparación con la CTU particular. En este ejemplo, el codificador de vídeo puede tener acceso a las CU contiguas espacialmente a partir de las cuales el codificador de vídeo puede obtener contexto para codificar la CTU particular. Sin embargo, el codificador de vídeo puede no haber codificado aún el encabezado de fragmento correspondiente con la CTU particular, y por lo tanto puede no ser capaz de codificar la CTU particular hasta que se codifique el encabezado de fragmento. En consecuencia, puede ser necesario que el codificador de vídeo codifique bloques adicionales del frente de onda anterior (es decir, hasta que se codifique el encabezado de fragmento), antes de comenzar a codificar la CTU particular. En este escenario, el codificador de vídeo no puede aprovechar las ventajas de WPP, tales como codificar la CTU particular en paralelo con una CTU posicionada por encima a la derecha.

[52] En lugar de permitir que un fragmento cruce un límite de frente de onda cuando el fragmento comienza en el medio del frente de onda, un codificador de vídeo puede implementar las técnicas de esta divulgación para restringir el proceso de codificación de tal manera que cuando un fragmento comienza en cualquier punto después del comienzo (es decir, la primera CTU) de un frente de onda, el fragmento finaliza dentro de ese frente de onda. Únicamente con fines de análisis, cualquier punto después del comienzo de un frente de onda generalmente se denomina de manera genérica en el presente documento el "medio" del frente de onda. Es decir, el "medio" del frente de onda como se usa en el presente documento no es necesariamente el punto medio, sino cualquier CTU (o cualquier bloque) de un frente de onda distinto del primer bloque ordinal del frente de onda. También se puede decir que dicho fragmento comienza "dentro" del frente de onda.

[53] Por ejemplo, el codificador de vídeo 20 puede determinar que un encabezado de fragmento ocurre en el medio de un frente de onda, y que el fragmento incluye todas las CTU restantes del frente de onda, así como al menos una CTU del frente de onda posicionado inmediatamente por debajo. En respuesta, el codificador de vídeo 20 puede insertar un símbolo de fin de fragmento al finalizar la codificación por entropía de una CTU hasta o incluyendo la última CTU del frente de onda. Es decir, el codificador de vídeo 20 puede garantizar que dicho fragmento finaliza dentro del frente de onda en el que comienza el fragmento, de tal manera que el fragmento no cruza los límites del frente de onda cuando el fragmento comienza en un bloque distinto del primer bloque ordinal del frente de onda. Además, el codificador de vídeo 20 puede insertar un encabezado de fragmento después del símbolo de fin de fragmento, indicando así que el siguiente frente de onda (por ejemplo, el frente de onda posicionado inmediatamente por debajo) corresponde al comienzo de un nuevo fragmento codificado. De manera similar, cuando se descodifica por entropía una imagen de acuerdo con WPP, el descodificador de vídeo 30 puede leer los encabezados de fragmento y/o los símbolos de fin de fragmento de una unidad de NAL recibida para

determinar que un fragmento codificado comienza en el medio de un frente de onda, y que el fragmento también finaliza dentro del mismo frente de onda en el que comienza el fragmento. En algunos ejemplos, el codificador de vídeo 20 puede determinar que dos o más fragmentos comienzan en el medio de un único frente de onda. En dichos ejemplos, el codificador de vídeo 20 puede determinar si el dicho último fragmento se desborda en un frente de onda siguiente, e implementar las restricciones descritas en el presente documento con respecto al dicho último frente de onda.

[54] De esta manera, se puede imponer una restricción de que para cualquier fragmento que comience en una CTU, u otro bloque, distinta de la primera CTU ordinal de un frente de onda, el fragmento finalizará dentro del frente de onda. Implementando estas restricciones, un codificador de vídeo, tal como el codificador de vídeo 20 y/o el descodificador de vídeo 30, puede mejorar la eficiencia implementando WPP. Más específicamente, el codificador de vídeo puede implementar las restricciones para garantizar que, mientras codifica una CTU de un frente de onda actual, el codificador de vídeo tiene acceso a cualesquiera datos de frentes de onda anteriores que pueden ser necesarios para codificar la CTU actual. Es decir, se puede garantizar que los datos de encabezado de fragmento para un fragmento que incluye una CTU actual estarán disponibles cuando se codifiquen por entropía uno o más elementos sintácticos de la CTU actual, de tal manera que el codificador de vídeo pueda determinar el contexto para codificar por entropía los elementos sintácticos correctamente.

[55] En algunos ejemplos, un fragmento puede comenzar en la primera CTU de un primer frente de onda, y cruzar el límite hasta un segundo frente de onda posicionado inmediatamente por debajo del primer frente de onda. En dichos ejemplos, el fragmento puede incluir múltiples CTU del segundo frente de onda, pero puede terminar dentro del segundo frente de onda. En otras palabras, el segundo frente de onda puede incluir CTU adicionales que pertenecen a un segundo fragmento diferente.

[56] En este ejemplo, mientras codifica una CTU del fragmento que está posicionado en el segundo frente de onda, el codificador de vídeo puede tener acceso a todos los datos del primer frente de onda que son necesarios para el proceso de codificación. Es decir, el codificador de vídeo ya habrá codificado los datos de encabezado de fragmento durante la codificación del frente de onda anterior y, por lo tanto, todavía puede estar permitido que un fragmento que comienza en la primera CTU ordinal de un frente de onda cruce el límite de frente de onda hasta un frente de onda siguiente. De manera adicional, utilizando las restricciones descritas en el presente documento, el codificador de vídeo puede garantizar que el segundo fragmento termina dentro del segundo frente de onda. Por ejemplo, si el codificador de vídeo determina que el segundo fragmento comienza en el medio del segundo frente de onda y por lo tanto finaliza con la última CTU del segundo frente de onda, el codificador de vídeo puede finalizar la codificación del segundo frente de onda de manera síncrona con la finalización de la codificación del segundo fragmento. En consecuencia, el comienzo de un tercer frente de onda, por definición, coincidiría con el comienzo de un tercer fragmento. Más específicamente, la primera CTU (más a la izquierda) del tercer frente de onda representaría la primera CTU del tercer fragmento. Si el tercer fragmento cruza el límite hasta el cuarto (o mayor) frente de onda, el codificador de vídeo puede tener acceso coherente a datos críticos de codificación de partes del tercer fragmento posicionadas en frentes de onda anteriores, mejorando así la capacidad del codificador de vídeo para realizar WPP. De esta manera, un codificador de vídeo puede implementar las técnicas de esta divulgación para codificar un fragmento actual de tal manera que, mientras se implementa WPP, los fragmentos siguientes se codifican de manera eficiente de acuerdo con WPP.

[57] Un codificador de vídeo, tal como un codificador de vídeo 20 y/o un descodificador de vídeo 30, puede activar la(s) restricción(es) basándose en si el codificador de vídeo está implementando actualmente WPP o no. Como un ejemplo, el descodificador de vídeo 30 puede determinar si WPP está habilitado actualmente utilizando datos sintácticos del flujo de bits indicativos de si WPP está habilitado. De forma similar, el codificador de vídeo 20 puede codificar datos sintácticos que representan si WPP está habilitado. Dichos datos sintácticos pueden codificarse en un conjunto de parámetros de vídeo (VPS), un conjunto de parámetros de secuencia (SPS), un conjunto de parámetros de imagen (PPS), mensajes de información de mejora suplementaria (SEI) o similares. En este ejemplo, en respuesta a determinar que WPP está habilitado, el codificador de vídeo 20 puede codificar por entropía, y el descodificador de vídeo 30 puede descodificar por entropía, una imagen utilizando WPP, al tiempo que observa las restricciones fragmento-frente de onda descritas anteriormente. En algunas implementaciones, un codificador de vídeo puede habilitar WPP, por ejemplo, en respuesta a determinar que WPP está deshabilitado actualmente.

[58] Además, el codificador de vídeo 20 puede enviar datos sintácticos, tales como datos sintácticos basados en bloques, datos sintácticos basados en tramas y datos sintácticos basados en GOP, al descodificador de vídeo 30, por ejemplo, en un encabezado de trama, un encabezado de bloque, un encabezado de fragmento o un encabezado de GOP. Los datos sintácticos de GOP pueden describir un cierto número de tramas en el respectivo GOP, y los datos sintácticos de trama pueden indicar un modo de codificación/predicción utilizado para codificar la trama correspondiente.

[59] Cada uno del codificador de vídeo 20 y el descodificador de vídeo 30 pueden implementarse como cualquiera entre una amplia variedad de circuitos codificadores o descodificadores adecuados, según corresponda, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos

de la aplicación (ASIC), matrices de puertas programables por campo (FPGA), circuitos de lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. Cada uno del codificador de vídeo 20 y 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 (CODEC) 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.

[60] La FIG. 2 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo 20 que puede implementar técnicas para codificar frentes de onda en paralelo. El codificador de vídeo 20 puede realizar la intra-codificación y la inter-codificación de bloques de vídeo dentro de fragmentos de vídeo. La intra-codificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial en el vídeo dentro de una trama o imagen de vídeo dada. La inter-codificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal en el vídeo dentro de tramas o imágenes adyacentes de una secuencia de vídeo. El intra-modo (modo I) puede referirse a cualquiera de varios modos de codificación de base espacial. Los inter-modos, tales como la predicción unidireccional (modo P) o la bi-predicción (modo B), pueden referirse a cualquiera de varios modos de codificación de base temporal.

[61] Como se muestra en la FIG. 2, el codificador de vídeo 20 recibe un bloque de vídeo actual dentro de una trama de vídeo a codificar. En el ejemplo de la FIG. 2, el codificador de vídeo 20 incluye una unidad de selección de modo 40, una memoria de imágenes de referencia 64, un sumador 50, una unidad de procesamiento de transformación 52, una unidad de cuantificación 54 y una unidad de codificación por entropía 56. A su vez, la unidad de selección de modo 40 incluye una unidad de compensación de movimiento 44, una unidad de estimación de movimiento 42, una unidad de intra-predicción 46 y una unidad de división 48. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 incluye además la unidad de cuantificación inversa 58, la unidad de transformación inversa 60 y el sumador 62. También puede incluirse un filtro de desbloqueo (no se muestra en la FIG. 2) para filtrar límites de bloque, para eliminar distorsiones de efecto pixelado del vídeo reconstruido. Si se desea, el filtro de desbloqueo filtrará habitualmente la salida del sumador 62. También pueden usarse filtros adicionales (en bucle o post-bucle), además del filtro de desbloqueo. Dichos filtros no se muestran por razones de brevedad pero, si se desea, pueden filtrar la salida del sumador 50 (tal como un filtro en bucle).

[62] 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 estar divididos 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 inter-predictiva del bloque de vídeo recibido con respecto a uno o más bloques en una o más tramas de referencia, para proporcionar la predicción temporal. La unidad de intra-predicción 46, de forma alternativa, puede llevar a cabo la codificación intra-predictiva del bloque de vídeo recibido, con respecto a uno o más bloques contiguos de la misma trama o fragmento que el bloque que va a codificarse, para proporcionar la predicción espacial. El codificador de vídeo 20 puede llevar a cabo múltiples pasadas de codificación, por ejemplo, para seleccionar un modo de codificación adecuado para cada bloque de datos de vídeo.

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

[64] La unidad de selección de modo 40 puede seleccionar uno de los modos de codificación, intra o inter, por ejemplo, en función de los resultados de errores, y proporciona el bloque intra-codificado o inter-codificado resultante al sumador 50 para generar datos de bloque residuales y al sumador 62 para reconstruir el bloque codificado para su uso como una trama de referencia. La unidad de selección de modo 40 proporciona además elementos sintácticos, tales como vectores de movimiento, indicadores de intra-modo, información de división y otra información sintáctica de este tipo, a la unidad de codificación por entropía 56.

[65] La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden estar altamente integradas, pero se ilustran por separado con fines conceptuales. La estimación de movimiento, realizada por la unidad de estimación de movimiento 42, es el proceso de generación de vectores de movimiento, que estiman el movimiento de 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 que puede determinarse mediante la suma de una diferencia absoluta (SAD), una suma de diferencia de cuadrados (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 de píxel fraccionarias de la imagen de referencia. Por lo tanto, la unidad de estimación de movimiento 42 puede realizar una búsqueda de movimiento en relación con las posiciones de píxeles completas y las posiciones de píxeles fraccionarias, y generar un vector de movimiento con una precisión de píxel fraccionaria.

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

[67] 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 diferencia de píxel, como se analiza posteriormente. 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 modo 40 también puede generar elementos sintácticos asociados con los bloques de vídeo y el fragmento de vídeo para su uso por el descodificador de vídeo 30 para descodificar los bloques de vídeo del fragmento de vídeo.

[68] La unidad de intra-predicción 46 puede realizar la intra-predicción de un bloque actual, como alternativa a la inter-predicción 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 intra-predicción 46 puede determinar un modo de intra-predicción a utilizar para codificar un bloque actual. En algunos ejemplos, la unidad de intra-predicción 46 puede codificar un bloque actual usando varios modos de intra-predicción, por ejemplo, durante diferentes pasadas de codificación, y la unidad de intra-predicción 46 (o la unidad de selección de modo 40, en algunos ejemplos) puede seleccionar un modo de intra-predicción adecuado a utilizar a partir de los modos probados.

[69] Por ejemplo, la unidad de intra-predicción 46 puede calcular valores de velocidad-distorsión usando un análisis de velocidad-distorsión para los diversos modos de intra-predicción probados, y seleccionar el modo de intra-predicción que tenga las mejores características de velocidad-distorsión entre los modos probados. El análisis de velocidad-distorsión determina en general una magnitud de distorsión (o de errores) entre un bloque codificado y un bloque original, no codificado, que se codificó para generar el bloque codificado, así como una velocidad binaria (es decir, un número de bits) usada para generar el bloque codificado. La unidad de intra-predicción 46 puede calcular relaciones proporcionales a partir de las distorsiones y velocidades para los diversos bloques codificados, para determinar qué modo de intra-predicción presenta el mejor valor de velocidad-distorsión para el bloque.

[70] Después de seleccionar un modo de intra-predicción para un bloque, la unidad de intra-predicción 46 puede proporcionar información, indicativa del modo de intra-predicción seleccionado para el bloque, a la unidad de codificación por entropía 56. La unidad de codificación por entropía 56 puede codificar la información indicando el modo de intra-predicción seleccionado. El codificador de vídeo 20 puede incluir datos de configuración en el flujo de bits transmitido, que pueden incluir una pluralidad de tablas de índices de modos de intra-predicción y una pluralidad de tablas de índices de modos de intra-predicción modificadas (también denominadas tablas de asignación de palabras de código), definiciones de contextos de codificación para varios bloques e indicaciones del modo de intra-predicción más probable, una tabla de índices de modos de intra-predicción y una tabla de índices de modos de intra-predicción modificadas a utilizar para cada uno de los contextos.

[71] El codificador de vídeo 20 forma un bloque de vídeo residual restando los datos de predicción de la unidad de selección de modo 40 del bloque de vídeo original que se está codificando. 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 del coseno (DCT) o una transformación conceptualmente similar, al bloque residual, generando 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 podrían usarse transformaciones de onduladas, transformaciones de enteros, transformaciones de subbandas u otros tipos de transformaciones. En cualquier caso, la unidad de procesamiento de transformación 52 aplica la transformación al bloque residual,

generando 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 transformaciones, tal como un dominio de frecuencia. La unidad de procesamiento de transformación 52 puede enviar los coeficientes de transformación resultantes a la unidad de cuantificación 54. La unidad de cuantificación 54 cuantifica los coeficientes de transformación para reducir adicionalmente la velocidad binaria. El proceso de cuantificación puede reducir la profundidad de bits asociada con algunos o la totalidad de los coeficientes. El grado de cuantificación puede modificarse ajustando un parámetro de cuantificación. En algunos ejemplos, la unidad de cuantificación 54 puede realizar, a continuación, una exploración de la matriz que incluye los coeficientes de transformación cuantificados. De forma alternativa, la unidad de codificación por entropía 56 puede llevar a cabo la exploración.

[72] Tras la cuantificación, la unidad de codificación por entropía 56 codifica por entropía los coeficientes de transformación cuantificados. Por ejemplo, la unidad de codificación por entropía 56 puede llevar a cabo la codificación de longitud variable adaptable al contexto (CAVLC), la codificación aritmética binaria adaptable al contexto (CABAC), la codificación aritmética binaria adaptable al contexto y basada en sintaxis (SBAC), la codificación por entropía mediante la división en intervalos de probabilidades (PIPE) u otra técnica de codificación por entropía. En el caso de la codificación por entropía basada en el contexto, el contexto puede basarse en bloques contiguos. Tras la codificación por entropía realizada por la unidad de codificación por entropía 56, el flujo de bits codificado puede transmitirse a otro dispositivo (por ejemplo, el descodificador de vídeo 30) o archivarse para su posterior transmisión o recuperación.

[73] En ejemplos, la unidad de codificación por entropía 56 puede codificar por entropía los coeficientes de transformación cuantificados utilizando procesamiento paralelo de frente de onda (WPP). Como se ha descrito con respecto a la FIG. 1, WPP puede incluir codificación por entropía de los coeficientes de transformación cuantificados en paralelo. Por ejemplo, la unidad de codificación por entropía 56 puede disponer los coeficientes de transformación cuantificados en múltiples filas, o frentes de onda. A su vez, la unidad de codificación por entropía 56 puede codificar cada coeficiente utilizando elementos sintácticos recibidos de la unidad de selección de modo 40, tal como uno o más vectores de movimiento asociados con coeficientes contiguos. En términos de codificación utilizando WPP, la unidad de codificación por entropía 56 puede, para un coeficiente en cualquier posición distinta del comienzo o el final de un frente de onda, utilizar vectores de movimiento asociados con coeficientes que están posicionados a la izquierda, por encima a la izquierda, por encima y por encima a la derecha del coeficiente a codificar.

[74] Para mejorar la eficiencia de la codificación por entropía basada en WPP, la unidad de codificación por entropía 56 puede implementar técnicas de esta divulgación para restringir la interacción fragmento-frente de onda tradicional con respecto a los coeficientes de transformación cuantificados. Como se describe, un codificador de vídeo, tal como el codificador de vídeo 20, puede dividir una imagen, o datos representativos de la imagen, en múltiples fragmentos. En términos de la FIG. 2, el flujo de coeficientes de transformación cuantificados se puede dividir en múltiples fragmentos. A su vez, un fragmento puede cubrir partes variables de uno o más frentes de onda delimitados por la unidad de codificación por entropía 56. Por ejemplo, un primer fragmento puede cubrir la totalidad de un primer frente de onda y una parte incompleta de un segundo frente de onda. Un segundo fragmento puede cubrir el resto del segundo frente de onda no cubierto por el primer fragmento, y una parte incompleta de un tercer frente de onda. De esta manera, la interacción fragmento-frente de onda tradicional proporcionada por WPP puede no correlacionar los puntos de inicio/fin de un fragmento con los de un frente de onda.

[75] Para mejorar la eficiencia de la codificación por entropía de acuerdo con WPP, la unidad de codificación por entropía 56 puede implementar una o más técnicas de esta divulgación. Por ejemplo, la unidad de codificación por entropía 56 puede determinar, basándose en los coeficientes de transformación cuantificados recibidos de la unidad de cuantificación 54, que un fragmento de una imagen de datos de vídeo comienza en un frente de onda, es decir, una fila de unidades de árbol de codificación (CTU), en la imagen en una posición distinta del comienzo del frente de onda. Basándose en la determinación, la unidad de codificación por entropía 56 puede determinar que el fragmento finaliza dentro del frente de onda, y codificar el fragmento basándose en la determinación. Más específicamente, la unidad de codificación por entropía 56 puede terminar el fragmento en el último coeficiente del frente de onda, tal como insertando un símbolo de fin de fragmento en una unidad de NAL después de codificar el último coeficiente del frente de onda. De esta manera, la unidad de codificación por entropía 56 puede garantizar que, mientras codifica una CTU particular, la unidad de codificación por entropía 56 tiene acceso a toda la información para codificar la CTU particular de acuerdo con WPP, y que los datos de encabezado de fragmento para la CTU particular ya se han codificado por entropía.

[76] Además, la unidad de codificación por entropía 56 puede insertar un encabezado de fragmento en la unidad de NAL antes de codificar el primer coeficiente del siguiente frente de onda. En este caso, la unidad de codificación por entropía 56 puede codificar el flujo de coeficientes de transformación cuantificados de tal manera que el comienzo del siguiente frente de onda coincida con el comienzo de un fragmento independiente. Si el nuevo fragmento abarca la totalidad del segundo frente de onda y se desborda en un tercer frente de onda, la unidad de codificación por entropía 56 puede tener acceso a todos los datos necesarios para codificar de manera eficiente el tercer frente de onda de acuerdo con WPP. Más específicamente, la unidad de codificación por entropía 56 puede

garantizar que el encabezado de fragmento para todas las CTU del tercer frente de onda se han codificado por entropía antes de que cualquier CTU del tercer frente de onda se vaya a codificar por entropía.

[77] En un ejemplo específico, mientras codifica el segundo coeficiente del tercer frente de onda, la unidad de codificación por entropía 56 puede acceder, a partir de los elementos sintácticos enviados por la unidad de selección de modo 40, a vectores de movimiento que identifican el primer coeficiente de cada uno del segundo y tercer frentes de onda (es decir, los coeficientes a la izquierda y por encima a la izquierda del coeficiente actual), el segundo coeficiente del segundo frente de onda (es decir, el coeficiente posicionado inmediatamente por encima del coeficiente actual) y el tercer coeficiente del segundo frente de onda (es decir, el coeficiente posicionado por encima a la derecha del coeficiente actual). Además, el encabezado de fragmento para el segundo coeficiente ya se ha codificado por entropía, ya que el encabezado de fragmento coincide con el primer coeficiente del segundo frente de onda. De esta manera, la unidad de codificación por entropía 56 puede implementar las técnicas de esta divulgación para codificar un fragmento actual de tal manera que la codificación de un fragmento siguiente utilizando WPP sea más eficiente.

[78] La unidad de cuantificación inversa 58 y la unidad de transformación inversa 60 aplican la cuantificación inversa y la transformación inversa, respectivamente, para reconstruir el bloque residual en el dominio de píxeles, por ejemplo, para su uso posterior como un bloque de referencia. La unidad de compensación de movimiento 44 puede calcular un bloque de referencia añadiendo el bloque residual a un bloque predictivo de una de las tramas de la 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 de píxeles fraccionarios para su uso en la estimación de movimiento. El sumador 62 añade el bloque residual reconstruido al bloque de predicción compensado por movimiento, generado por la unidad de compensación de movimiento 44, para generar un bloque de vídeo reconstruido para su almacenamiento en la memoria de imágenes de referencia 64. El bloque de vídeo reconstruido puede utilizarse por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 como un bloque de referencia para inter-codificar un bloque en una trama de vídeo siguiente.

[79] Como se ha descrito anteriormente, el codificador de vídeo 20 representa un ejemplo de un codificador de vídeo configurado para determinar que un fragmento de una imagen de datos de vídeo comienza en una fila de unidades de árbol de codificación (CTU) en la imagen en una posición distinta del comienzo de la fila, basándose en la determinación, determinar que el fragmento finaliza dentro de la fila de CTU, y codificar el fragmento basándose en la determinación de que el fragmento finaliza dentro de la fila de CTU. En ejemplos, el codificador de vídeo 20 puede estar incluido en un dispositivo para codificar datos de vídeo, tal como un ordenador de sobremesa, un ordenador plegable (es decir, portátil), un ordenador de tableta, un descodificador, un equipo telefónico portátil tal como un denominado teléfono "inteligente", un denominado panel "inteligente", un televisor, una cámara, un dispositivo de visualización, un reproductor de medios digitales, una consola de videojuegos, un dispositivo de flujos de transmisión continua de vídeo o similares. En ejemplos, un dispositivo de este tipo para la codificación de datos de vídeo puede incluir uno o más de un circuito integrado, un microprocesador y un dispositivo de comunicación que incluye el codificador de vídeo 20.

[80] La FIG. 3 es un diagrama de bloques que ilustra un ejemplo de descodificador de vídeo 30 que puede implementar las técnicas para codificar frentes de onda en paralelo. En el ejemplo de la FIG. 3, el descodificador de vídeo 30 incluye una unidad de descodificación por entropía 70, una unidad de compensación de movimiento 72, una unidad de intra-predicción 74, una unidad de cuantificación inversa 76, una unidad de transformación inversa 78, una memoria de imágenes de referencia 82 y un sumador 80. En algunos ejemplos, el descodificador de vídeo 30 puede llevar a cabo una pasada de descodificación generalmente recíproca a la pasada de codificación descrita con respecto al codificador de vídeo 20 (FIG. 2). La unidad de compensación de movimiento 72 puede generar datos de predicción basándose en vectores de movimiento recibidos desde la unidad de descodificación por entropía 70, mientras que la unidad de intra-predicción 74 puede generar datos de predicción basándose en indicadores de modo de intra-predicción recibidos de la unidad de descodificación por entropía 70.

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

[82] La unidad de descodificación por entropía 70 puede implementar las técnicas de esta divulgación para restringir la interacción fragmento-frente de onda para descodificar por entropía de manera más eficiente una imagen utilizando procesamiento paralelo de frente de onda (WPP). Por ejemplo, la unidad de descodificación por entropía 70 puede determinar que un fragmento comienza en el medio de un frente de onda, tal como determinando que un encabezado de fragmento en una unidad de NAL recibida coincide con una CTU que no es la primera CTU de su respectivo frente de onda. Basándose en la determinación, la unidad de descodificación por entropía 70 puede

determinar que el fragmento finaliza dentro del mismo frente de onda, por ejemplo, determinando que la unidad de NAL recibida incluye un símbolo de fin de fragmento al final de la última CTU del frente de onda actual.

[83] Restringiendo la interacción fragmento-frente de onda de esta manera, la unidad de descodificación por entropía 70 puede descodificar por entropía una imagen de manera más eficiente utilizando WPP. Por ejemplo, la unidad de descodificación por entropía 70 puede garantizar que, mientras descodifica una CTU particular, la unidad de descodificación por entropía 70 tiene acceso a toda la información necesaria para descodificar la CTU particular utilizando WPP, y que el encabezado de fragmento para la CTU particular ya se ha descodificado por entropía en el instante en que la unidad de descodificación por entropía 70 está lista para descodificar la CTU particular. De esta manera, la unidad de descodificación por entropía 70 puede implementar las técnicas de esta divulgación para descodificar de manera más eficiente una imagen de acuerdo con WPP.

[84] Cuando el fragmento de vídeo se codifica como un fragmento intra-codificado (I), la unidad de intra-predicción 74 puede generar datos de predicción para un bloque de vídeo del fragmento de vídeo actual basándose en un modo de intra-predicción señalado y datos de bloques previamente descodificados de la trama o imagen actual. Cuando la trama de vídeo se codifica como un fragmento inter-codificado (es decir, B, P o GPB), la unidad de compensación de movimiento 72 genera bloques predictivos para un bloque de vídeo del fragmento de vídeo actual basándose en los vectores de movimiento y en otros elementos sintácticos recibidos de la unidad de descodificación por entropía 70. Los bloques predictivos pueden generarse a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El descodificador de vídeo 30 puede construir las listas de tramas de referencia, la Lista 0 y la Lista 1, usando técnicas de construcción predeterminadas basándose en las imágenes de referencia almacenadas en la memoria de imágenes de referencia 82. La unidad de compensación de movimiento 72 determina la información de predicción para un bloque de vídeo del fragmento de vídeo actual analizando los vectores de movimiento y otros elementos sintácticos, y usa la información de predicción para generar los bloques predictivos del bloque de vídeo actual que se está descodificando. Por ejemplo, la unidad de compensación de movimiento 72 usa algunos de los elementos sintácticos recibidos para determinar un modo de predicción (por ejemplo, intra-predicción o inter-predicción) usado para codificar los bloques de vídeo del fragmento de vídeo, un tipo de fragmento de inter-predicción (por ejemplo, fragmento B, fragmento P o fragmento GPB), información de construcción para una o más de las listas de imágenes de referencia del fragmento, vectores de movimiento para cada bloque de vídeo inter-codificado del fragmento, el estado de inter-predicción para cada bloque de vídeo inter-codificado del fragmento y otra información para descodificar los bloques de vídeo en el fragmento de vídeo actual.

[85] 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 bloques de referencia. En este caso, la unidad de compensación de movimiento 72 puede determinar los filtros de interpolación utilizados por el codificador de vídeo 20 a partir de los elementos sintácticos recibidos y utilizar los filtros de interpolación para generar bloques predictivos.

[86] La unidad de cuantificación inversa 76 cuantifica de manera inversa, es decir, descuantifica, los coeficientes de transformación cuantificados proporcionados en el flujo de bits y descodificados por la unidad de descodificación por entropía 80. El proceso de cuantificación inversa puede incluir el uso de un parámetro de cuantificación QPy, calculado por el descodificador de vídeo 30 de cada bloque de vídeo en el fragmento de vídeo para determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa que debería aplicarse.

[87] La unidad 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 producir bloques residuales en el dominio de píxeles.

[88] Después de que la unidad de compensación de movimiento 72 genera el bloque predictivo para el bloque de vídeo actual basándose en los vectores de movimiento y otros elementos sintácticos, el descodificador de vídeo 30 forma un bloque de vídeo descodificado sumando los bloques residuales de la unidad de transformación inversa 78 y los correspondientes bloques predictivos generados por la unidad de compensación de movimiento 72. El sumador 80 representa el componente o los componentes que llevan a cabo esta operación de suma. Si se desea, también puede aplicarse un filtro de desbloqueo para filtrar los bloques descodificados con el fin de eliminar distorsiones de efecto pixelado. También pueden utilizarse otros filtros de bucle (ya sea en el bucle de codificación o después del bucle de codificación) para suavizar las transiciones de píxeles o mejorar de otro modo la calidad del vídeo. Los bloques de vídeo descodificados en una trama o imagen dada son a continuación almacenados en la memoria de imágenes de referencia 82, que almacena imágenes de referencia usadas para la posterior compensación de movimiento. La memoria de imágenes de referencia 82 almacena también vídeo descodificado para su presentación posterior en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la FIG. 1.

[89] De esta manera, el descodificador de vídeo 30 representa un ejemplo de un codificador de vídeo configurado para determinar que un fragmento de una imagen de datos de vídeo comienza en una fila de unidades de árbol de codificación (CTU) en la imagen en una posición distinta del comienzo de la fila, basándose en la

determinación, determinar que el fragmento finaliza dentro de la fila de CTU, y codificar el fragmento basándose en la determinación de que el fragmento finaliza dentro de la fila de CTU. En ejemplos, el descodificador de vídeo 30 puede estar incluido en un dispositivo para codificar datos de vídeo, tal como un ordenador de sobremesa, un ordenador plegable (es decir, portátil), un ordenador tipo tableta, un descodificador, un equipo telefónico portátil tal como un denominado teléfono "inteligente", un denominado panel "inteligente", un televisor, una cámara, un dispositivo de visualización, un reproductor de medios digitales, una consola de videojuegos, un dispositivo de transmisión continua de vídeo o similares. En ejemplos, un dispositivo de este tipo para la codificación de datos de vídeo puede incluir uno o más entre un circuito integrado, un microprocesador y un dispositivo de comunicación que incluye el descodificador de vídeo 30.

[90] La FIG. 4 es un diagrama conceptual que ilustra una imagen de ejemplo 100 dividida en frentes de onda 150-160. Cada uno de los frentes de onda 150-160 incluye un número de bloques. Debe observarse que la imagen 100 puede incluir frentes de onda adicionales, y que cada frente de onda puede incluir bloques adicionales a los mostrados. Cada uno de los bloques puede corresponder, por ejemplo, a una CTU.

[91] Un codificador de vídeo, tal como el codificador de vídeo 20 o el descodificador de vídeo 30, puede configurarse para codificar frentes de onda 150-160 en paralelo. El codificador de vídeo 20 puede comenzar a codificar un frente de onda después de que se hayan codificado dos bloques del frente de onda por encima. La FIG. 4 ilustra los bloques después de los cuales se puede codificar un frente de onda utilizando puntos negros conectados por una flecha curvada relativamente horizontal. Por ejemplo, el bloque 134 del frente de onda 156 puede codificarse después de que se haya codificado el bloque 128 del frente de onda 154. Como un ejemplo, un codificador de vídeo puede codificar cada uno de los bloques marcados con una "X", es decir, los bloques 116, 124, 132 y 136, en paralelo. Las líneas discontinuas 102, 104, 106 y 108 representan bloques que se han analizado y de los que se dispone información para su recuperación en un instante de codificación particular, en el ejemplo de la FIG. 4. El instante de codificación particular puede corresponder al instante en el que se codifican los bloques marcados con una "X", es decir, los bloques 116, 124, 132 y 136.

[92] En consecuencia, el codificador de vídeo puede recuperar información de contexto para un bloque marcado con una "X" a partir de los bloques apuntados por flechas blancas continuas en la FIG. 4. Como se muestra en la FIG. 4, cada uno de los bloques a los que apunta una flecha blanca continua está dentro de una de las líneas discontinuas 102, 104, 106 y 108. Por ejemplo, un codificador de vídeo puede recuperar información de contexto para el bloque 116 a partir del bloque 114; el codificador de vídeo puede recuperar información de contexto para el bloque 124 a partir de los bloques 110, 112, 114 y/o 122; el codificador de vídeo puede recuperar información de contexto para el bloque 132 a partir de los bloques 118, 120, 122 y/o 130; y el codificador de vídeo puede recuperar información de contexto para el bloque 136 a partir de los bloques 126, 128, 130 y/o 134. De manera colectiva, las líneas discontinuas 102, 104, 106 y 108 incluyen un grupo de CTU, es decir, una recopilación de bloques de imagen 100 que el codificador de vídeo es capaz de codificar en un instante dado de acuerdo con WPP.

[93] De acuerdo con las técnicas de esta divulgación, cada uno de los frentes de onda 150-160 puede incluir partes o totalidades de uno o más fragmentos. De manera alternativa, un fragmento puede incluir partes o totalidades de uno o más frentes de onda, tales como uno o más de los frentes de onda 150-160. Por ejemplo, un primer fragmento puede incluir bloques de frentes de onda 150 y 152, un segundo fragmento puede incluir bloques de frentes de onda 152, 154 y 156, y un tercer fragmento puede incluir bloques de frentes de onda 158 y 160. De esta manera, cuando un fragmento cruza un límite entre dos frentes de onda, el fragmento puede incluir algunos o todos los bloques de los dos frentes de onda.

[94] Se supone, por ejemplo, que un codificador de vídeo va a codificar el bloque 132. Con el fin de inicializar un estado de contexto para el frente de onda 154, que incluye el bloque 132, un codificador de vídeo puede necesitar uno o más parámetros de un encabezado de fragmento para un fragmento que incluye el bloque 132. Si se permitiese que el fragmento comenzase en un bloque en el medio del frente de onda 152 y cruzase el límite entre el frente de onda 152 y el frente de onda 154, el codificador de vídeo podría tener que esperar para codificar el primer bloque en el fragmento para recuperar la información del encabezado de fragmento. Por ejemplo, si el fragmento comenzase en la posición horizontal del bloque en el frente de onda 152 por debajo del bloque 116, este bloque aún no se habría analizado y, por lo tanto, el codificador de vídeo tendría que esperar hasta que el bloque se analizase antes de que el codificador de vídeo pudiese comenzar a codificar el frente de onda 154. Sin embargo, el codificador de vídeo puede implementar técnicas de esta divulgación para proporcionar que, si un fragmento comienza en un frente de onda de la imagen 100 en una posición distinta del comienzo del frente de onda, entonces el fragmento finaliza dentro de ese frente de onda particular. En otras palabras, cualquier frente de onda de la imagen 100 puede o bien comenzar con un encabezado de fragmento, o bien finalizar con un símbolo de fin de fragmento (o ambos). Restringiendo la interacción fragmento-frente de onda de esta manera, el codificador de vídeo puede garantizar que, mientras codifica un bloque de la imagen particular 100, el codificador de vídeo tiene acceso a toda la información necesaria para codificar el bloque de acuerdo con WPP, y que el encabezado de fragmento correspondiente al bloque ya se ha codificado. Así pues, el codificador de vídeo puede evitar ocurrencias de situaciones que requieren que el codificador de vídeo espere para codificar un bloque de acuerdo con el procesamiento paralelo de frente de onda (WPP).

[95] Más específicamente, un codificador de vídeo puede implementar las técnicas para garantizar que todos los datos necesarios para codificar el bloque de acuerdo con el procesamiento paralelo de frente de onda (WPP), por ejemplo, utilizando CABAC, estén disponibles, y que el bloque está listo para codificarse en que el encabezado de fragmento para el bloque ya se ha codificado. Aunque diversos codificadores de vídeo pueden implementar las técnicas, con fines de explicación, una o más de las técnicas se describen con referencia al codificador de vídeo 20 de la FIG. 2 y el descodificador de vídeo 30 de la FIG. 3, y sus respectivos componentes. Por ejemplo, la unidad de descodificación por entropía 70 del descodificador de vídeo 30 puede restringir la interacción fragmento-frente de onda de la imagen 100 para garantizar que todos los datos necesarios para codificar un bloque de acuerdo con WPP están disponibles y que el encabezado de fragmento correspondiente para el bloque ya se ha codificado. Por ejemplo, la unidad de descodificación por entropía 70 puede determinar que un fragmento de la imagen 100 comienza en el frente de onda 150, pero en una posición distinta del comienzo del frente de onda 150 (por ejemplo, en el bloque 110). Basándose en la determinación, la unidad de descodificación por entropía 70 puede determinar que el fragmento finaliza dentro del frente de onda 150. Por ejemplo, la unidad de descodificación por entropía 70 puede detectar, en una unidad de NAL recibida que representa partes de la imagen 100, un símbolo de fin de fragmento inmediatamente después del último bloque codificado del frente de onda 150, y detectar un encabezado de fragmento inmediatamente antes del primer bloque codificado del frente de onda 152. De esta manera, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden garantizar que el encabezado de fragmento para cualquier bloque particular del frente de onda 152 se ha codificado antes de que el bloque particular esté listo para codificarse.

[96] En el contexto de codificación de la imagen 100 de acuerdo con WPP, la unidad de codificación por entropía 56 del codificador de vídeo 20 puede detectar que el fragmento que comienza en el bloque 110 también incluye uno o más bloques del frente de onda 152. Este concepto se denomina en el presente documento "desbordamiento de fragmento". En este caso, la unidad de codificación por entropía 56 puede terminar el fragmento después de incluir el último bloque del frente de onda 150, e iniciar un nuevo fragmento que incluye el primer bloque del frente de onda 152. Como se describe, la unidad de codificación por entropía 56 puede generar una unidad de NAL que incluye datos codificados que representan la imagen 100, e insertar un símbolo de fin de fragmento después de los datos que representan el último bloque del frente de onda 150 (como codificado). De manera similar, la unidad de codificación por entropía 56 puede insertar, en la misma o en la siguiente unidad de NAL, un encabezado de fragmento inmediatamente antes de los datos que representan el primer bloque de frente de onda 152 como codificado. Implementando estas restricciones, el codificador de vídeo 20 puede garantizar que el encabezado de fragmento para un bloque determinado de los frentes de onda 150 y/o 152 se ha codificado antes de que el bloque dado esté listo para codificarse. De esta manera, el codificador de vídeo 20 puede implementar técnicas de esta divulgación para restringir la interacción fragmento-frente de onda para utilizar WPP de manera más eficiente, y para permitir que el descodificador de vídeo 30 también utilice WPP de manera más eficiente.

[97] De esta manera, un codificador de vídeo puede determinar que un fragmento de la imagen 100 comienza en una fila de CTU en la imagen 100 distinta de un comienzo de la fila. Basándose en la determinación, el codificador de vídeo puede determinar que el fragmento finaliza dentro de la fila de CTU, y codificar el fragmento basándose en la determinación de que el fragmento finaliza dentro de la fila de CTU.

[98] Como se ha analizado anteriormente, en algunos ejemplos, un codificador de vídeo, tal como el codificador de vídeo 20 o el descodificador de vídeo 30, puede configurarse para codificar datos de vídeo de tal manera que, cuando un fragmento de los datos de vídeo incluye una primera fila de unidades de árbol de codificación (CTU) que comprende una fila completa, y una parte de una segunda fila de CTU, comprendiendo la segunda fila menos CTU que la fila completa, el fragmento comienza al comienzo de la al menos una fila completa de unidades de codificación de máximo tamaño. Por ejemplo, se supone que un fragmento comienza al comienzo del frente de onda 150. En este ejemplo, el fragmento puede finalizar en el medio de (es decir, antes del final de) un frente de onda siguiente, por ejemplo, el frente de onda 152.

[99] Se supone, por ejemplo, que el fragmento finaliza en el bloque 120. Esto estaría permitido por la restricción descrita anteriormente, porque el fragmento comienza al comienzo de un frente de onda, es decir, el frente de onda 150 en este ejemplo. Así pues, un fragmento siguiente puede comenzar en el bloque 122. Sin embargo, a este fragmento no se le permitiría cruzar el límite al final del frente de onda 152, en este ejemplo. Así pues, el fragmento finalizaría al final del frente de onda 152. Por supuesto, se pueden añadir fragmentos adicionales dentro del frente de onda 152, siempre que un fragmento no cruce el límite entre el frente de onda 152 y el frente de onda 154, en este ejemplo.

[100] La FIG. 5 es un diagrama de flujo que ilustra un proceso 180 de ejemplo mediante el cual un codificador de vídeo puede implementar técnicas de esta divulgación para codificar frentes de onda en paralelo. La FIG. 5 ilustra el proceso 180 de ejemplo mediante el cual un codificador de vídeo, tal como el codificador de vídeo 20, puede codificar una imagen, tal como una trama de datos de vídeo, utilizando una o más técnicas de esta divulgación. Si bien el proceso 180 puede realizarse mediante diversos dispositivos de acuerdo con los aspectos de esta divulgación, con fines de explicación, el proceso 180 se describe en el presente documento con respecto a los dispositivos de las FIG. 1-2 y sus respectivos componentes, así como la imagen 100 de la FIG. 4. El proceso 180

puede comenzar cuando un dispositivo recibe una imagen de datos de vídeo (182). Como un ejemplo, el dispositivo de origen 12 puede recibir la imagen 100 a través de uno o más dispositivos de entrada.

[101] De manera adicional, el dispositivo de origen 12 puede habilitar el procesamiento paralelo de frente de onda (WPP) (184). Por ejemplo, el dispositivo de origen 12 puede habilitar WPP, haciendo así que el codificador de vídeo 20 codifique la imagen 100 de acuerdo con WPP. Un codificador de vídeo, tal como el codificador de vídeo 20, puede determinar frentes de onda de la imagen 100 (186). Por ejemplo, el codificador de vídeo 20 puede determinar un número de bloques (por ejemplo, CTU) por frente de onda asociados con la codificación basada en WPP de la imagen 100, y determinar una transición de frente de onda después de alcanzar cada múltiplo entero del número de bloques.

[102] De manera adicional, el codificador de vídeo 20 puede determinar encabezados de fragmento para la imagen 100 (188). Más específicamente, el codificador de vídeo 20 puede utilizar encabezados de fragmento para indicar una transición de fragmento, es decir, el comienzo de un nuevo fragmento de la imagen 100. Por ejemplo, el codificador de vídeo 20 puede insertar un encabezado de fragmento en una parte particular de la imagen 100 que corresponde al comienzo de un nuevo fragmento. En algunos ejemplos, el codificador de vídeo 20 puede indicar la transición de fragmento basándose en un símbolo de fin de fragmento, tal como insertando un símbolo de fin de fragmento en una parte de la imagen 100 para indicar el final de un fragmento. En algunos ejemplos, el codificador de vídeo 20 puede indicar la transición de fragmento utilizando una secuencia de un símbolo de fin de fragmento seguido inmediatamente por un encabezado de fragmento, tal como insertando un símbolo de fin de fragmento para indicar el final de un fragmento, e insertando un encabezado de fragmento inmediatamente después del símbolo de fin de fragmento, para indicar el inicio de un nuevo fragmento.

[103] El codificador de vídeo puede determinar si el fragmento actual comienza después de la primera CTU de un frente de onda (190). Por ejemplo, el codificador de vídeo 20 puede determinar que el fragmento actual comienza después de la primera CTU del frente de onda 150 (por ejemplo, en el "medio" del frente de onda), si el codificador de vídeo 20 detecta, o inserta, un encabezado de fragmento en el bloque 110. En este ejemplo, si el codificador de vídeo 20 determina que el fragmento actual comienza después de la primera CTU del frente de onda (rama "Sí" de 190), entonces el codificador de vídeo 20 puede determinar que el fragmento actual finaliza dentro del frente de onda actual (192). Por ejemplo, el codificador de vídeo 20 puede determinar que el fragmento actual finaliza dentro del frente de onda 150 colocando un símbolo de fin de fragmento en la unidad de NAL generada antes de delimitar una transición al frente de onda 152. Determinando las transiciones de fragmento descritas, el codificador de vídeo 20 puede garantizar que el codificador de vídeo 20 (y/o el decodificador de vídeo 30) tiene acceso a toda la información necesaria para codificar un bloque del frente de onda 152, y que el encabezado de fragmento para el bloque ya se ha codificado.

[104] Por otro lado, si el codificador de vídeo determina que el fragmento actual no comienza después de la primera CTU de un frente de onda, es decir, el encabezado de fragmento coincide con la primera CTU del frente de onda (rama "NO" de 190), el codificador de vídeo puede continuar determinando los encabezados de fragmento para la imagen 100 (188). Por ejemplo, el codificador de vídeo 30 puede codificar encabezados de fragmento siguientes (y/o símbolos de fin de fragmento), basándose en determinar que el fragmento actual comienza al comienzo de un frente de onda. De esta manera, el codificador de vídeo 20 puede implementar el proceso 180 para restringir la interacción fragmento-frente de onda de tal manera que el codificador de vídeo 20 y el decodificador de vídeo 30 tienen acceso a todos los datos necesarios (incluyendo un encabezado de fragmento ya codificado) para codificar de manera eficiente una CTU actual, de acuerdo con WPP.

[105] La FIG. 6 es un diagrama de flujo que ilustra otro proceso 200 de ejemplo mediante el cual un codificador de vídeo puede decodificar una imagen codificada, tal como una trama de datos de vídeo, utilizando una o más técnicas de esta divulgación. Si bien el proceso 200 puede realizarse mediante diversos dispositivos de acuerdo con los aspectos de esta divulgación, con fines de explicación, el proceso 200 se describe en el presente documento con respecto a los dispositivos de las FIG. 1 y 3 y sus respectivos componentes, así como la imagen 100 de la FIG. 4. El proceso 200 puede comenzar cuando el decodificador de vídeo 30 recibe una imagen codificada de datos de vídeo (202). Como un ejemplo, el dispositivo de destino 14 puede recibir una versión codificada de la imagen 100 en la interfaz de entrada 128.

[106] De manera adicional, el decodificador de vídeo 30 puede permitir el procesamiento paralelo de frente de onda (WPP) (204). En algunos ejemplos, el medio legible por ordenador 16 puede incluir datos de señalización que indican que se debe habilitar WPP. Por el contrario, en otros ejemplos, el decodificador de vídeo 30 puede determinar una indicación implícita para habilitar WPP, basándose en si los datos de vídeo incluidos en el medio legible por ordenador 16 se ajustan a una norma particular y/o a un perfil particular de una norma. Por ejemplo, el dispositivo de destino 14 puede habilitar WPP para hacer que el decodificador de vídeo 30 decodifique la versión codificada recibida de la imagen 100 de acuerdo con WPP. De manera adicional, el decodificador de vídeo 30 puede determinar el inicio de un nuevo frente de onda de la imagen codificada 100 (206). Por ejemplo, el decodificador de vídeo 30 puede determinar que el primer bloque (por ejemplo, CTU) de la imagen codificada 100 indica el inicio de un nuevo frente de onda 150. Además, el decodificador de vídeo 30 puede determinar un número de CTU por frente de onda asociado con la codificación basada en WPP de la imagen 100, y determinar el inicio de

un nuevo frente de onda (o "transición de frente de onda") después de alcanzar cada múltiplo entero del número de CTU.

[107] El descodificador de vídeo 30 puede descodificar datos de vídeo del fragmento actual del frente de onda actual (por ejemplo, el frente de onda 150) de la versión codificada de la imagen 100 (208). Más específicamente, el descodificador de vídeo 30 puede descodificar el frente de onda 150 en una base por CTU, comenzando en la CTU más a la izquierda, descodificando después la siguiente CTU a la derecha, y así sucesivamente. De manera adicional, el descodificador de vídeo 30 puede determinar si el descodificador de vídeo 30 detecta o no un encabezado de fragmento antes de alcanzar el final (por ejemplo, la CTU más a la derecha) del frente de onda 150 (210). El descodificador de vídeo 30 puede determinar el inicio de un nuevo fragmento de la imagen codificada 100, o una "transición de fragmento", basándose en la detección de un encabezado de fragmento en la imagen codificada 100. Si el descodificador de vídeo 30 no detecta un encabezado de fragmento antes del final del frente de onda 150 (rama "NO" de 210), el descodificador de vídeo 30 puede detectar el inicio de un nuevo frente de onda 152 (206). Más específicamente, en este escenario, el descodificador de vídeo 30 puede determinar que al menos parte del frente de onda 152, que incluye la primera CTU del frente de onda 152, pertenece al mismo fragmento que las CTU del frente de onda 152.

[108] Por el contrario, si el descodificador de vídeo 30 detecta un encabezado de fragmento antes del final del frente de onda 150 (rama "Sí" de 210), el descodificador de vídeo 30 puede determinar si el encabezado de fragmento coincide con la primera CTU del frente de onda 150 (212). En otras palabras, el descodificador de vídeo 30 puede determinar si el fragmento actual comienza en la misma CTU que el frente de onda 150 actual. Si el descodificador de vídeo 30 determina que el encabezado de fragmento detectado coincide con la primera CTU del frente de onda 150 (rama "Sí" de 212), el descodificador de vídeo 30 puede continuar descodificando datos de vídeo del fragmento actual del frente de onda 150 (208).

[109] Por el contrario, si el descodificador de vídeo 30 determina que el encabezado de fragmento detectado no coincide con la primera CTU del frente de onda 150 (rama "NO" de 212), el descodificador de vídeo 30 puede determinar que el fragmento actual finaliza dentro del frente de onda 150 (214). Más específicamente, el descodificador de vídeo 30 puede determinar que el fragmento actual finaliza dentro del (por ejemplo, en o antes de la última CTU/CTU más a la derecha del) frente de onda 150 basándose en las restricciones de interacción fragmento-frente de onda habilitadas por las técnicas de esta divulgación. Además, basándose en determinar que el fragmento actual finaliza dentro del frente de onda 150, el descodificador de vídeo puede continuar descodificando datos de vídeo del fragmento actual del frente de onda 150.

[110] Restringiendo la interacción fragmento-frente de onda de la manera ilustrada en la FIG. 6, el descodificador de vídeo 30 puede garantizar que, mientras descodifica una CTU del fragmento que está posicionado en un frente de onda, tal como el frente de onda 152, el descodificador de vídeo 30 tiene acceso a todos los datos del frente de onda 150 que son necesarios para el proceso de descodificación del frente de onda 152. Es decir, el descodificador de vídeo 30 ya habrá o bien descodificado los datos del encabezado de fragmento para un fragmento en el frente de onda 152 durante la descodificación del frente de onda 150, o bien el encabezado de fragmento para el fragmento en el frente de onda 152 ocurrirá al comienzo del frente de onda 152, y por lo tanto, el descodificador de vídeo 30 puede comenzar a descodificar el frente de onda 152 basándose en tener acceso a todos los datos críticos de descodificación necesarios de acuerdo con WPP.

[111] Debe reconocerse que, según el ejemplo, ciertos actos o sucesos de cualquiera de las técnicas descritas 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 de las técnicas). Además, en ciertos ejemplos, los actos o sucesos pueden realizarse simultáneamente, por ejemplo, mediante el procesamiento de múltiples hilos, el procesamiento de interrupciones o múltiples procesadores, en lugar de secuencialmente.

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

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

[114] Las instrucciones pueden ser ejecutadas por uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), matrices lógicas programables por campo (FPGA) u otros circuitos lógicos integrados o discretos equivalentes. Por consiguiente, el término "procesador", como se usa en el presente documento, puede referirse a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento puede proporcionarse dentro de módulos de hardware y/o software dedicados configurados para la codificación y la decodificación, o incorporarse en un códec combinado. Además, las técnicas podrían implementarse completamente en uno o más circuitos o elementos lógicos.

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

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

REIVINDICACIONES

1. Un procedimiento de codificación de datos de vídeo de codificación de vídeo de alta eficiencia -HEVC-, comprendiendo los datos de vídeo una imagen (100) dividida en una pluralidad de filas (150-160) de unidades de árbol de codificación -CTU- estando la imagen también dividida en una pluralidad de fragmentos, incluyendo cada fragmento de la imagen una o más CTU consecutivas en orden de codificación, incluyendo cada fila de CTU de la imagen partes o totalidades de uno o más fragmentos, comprendiendo el procedimiento:

para todos los fragmentos de todas las imágenes (100) de los datos de vídeo, en respuesta a la determinación (190) de que un fragmento comienza en una fila (150-160) de CTU en la imagen en una posición distinta de un comienzo (126, 134) de la fila, codificar CTU del fragmento en orden de codificación sin cruzar a una fila posterior de CTU en la imagen antes de alcanzar el final del fragmento, siendo la primera CTU en orden de codificación de la fila posterior el comienzo de un fragmento diferente; y

codificar todos los fragmentos de todas las imágenes de los datos de vídeo de tal manera que todos los fragmentos que comienzan en una posición distinta de un comienzo de una fila correspondiente de CTU también finalizan dentro de la fila correspondiente de CTU,

en el que el orden de codificación de CTU es secuencialmente de izquierda a derecha en una fila de CTU y avanza a una fila posterior que sigue a la CTU más a la derecha de una fila,

en el que, para al menos un fragmento, si la primera CTU en orden de codificación del fragmento es la primera CTU en orden de codificación de una fila, entonces codificar el fragmento de tal manera que incluye una o más CTU de una o más filas posteriores de CTU.
2. El procedimiento según la reivindicación 1, que comprende además determinar que el fragmento finaliza en un final de la fila de CTU
3. El procedimiento según la reivindicación 1, que comprende además determinar que el fragmento finaliza antes del final de la fila de CTU.
4. Un dispositivo (12) para codificación de datos de vídeo de codificación de vídeo de alta eficiencia -HEVC-, comprendiendo los datos de vídeo una imagen (100) dividida en una pluralidad de filas (150-160) de unidades de árbol de codificación -CTU- estando la imagen también dividida en una pluralidad de fragmentos, incluyendo cada fragmento de la imagen una o más CTU consecutivas en orden de codificación, incluyendo cada fila de CTU de la imagen partes o totalidades de uno o más fragmentos, comprendiendo el dispositivo:

medios para, para todos los fragmentos de todas las imágenes (100) de los datos de vídeo, en respuesta a la determinación (190) de que un fragmento comienza en una fila (150-160) de CTU en la imagen en una posición distinta de un comienzo (126, 134) de la fila, codificar CTU del fragmento en orden de codificación sin cruzar a una fila posterior de CTU en la imagen antes de alcanzar el final del fragmento, siendo la primera CTU en orden de codificación de la fila posterior el comienzo de un fragmento diferente; y

medios para codificar todos los fragmentos de todas las imágenes de los datos de vídeo de tal manera que todos los fragmentos que comienzan en una posición distinta de un comienzo de una fila correspondiente de CTU también finalizan dentro de la fila correspondiente de CTU,

en el que el orden de codificación de CTU es secuencialmente de izquierda a derecha en una fila de CTU y avanza a una fila posterior que sigue a la CTU más a la derecha de una fila,

en el que, para al menos un fragmento, si la primera CTU en orden de codificación del fragmento es la primera CTU en orden de codificación de una fila, entonces codificar el fragmento de tal manera que incluye una o más CTU de una o más filas posteriores de CTU.
5. El dispositivo según la reivindicación 4, que comprende además medios para determinar que el fragmento finaliza en un final de la fila de CTU
6. El dispositivo, según la reivindicación 4, que comprende además medios para determinar que el fragmento finaliza antes del final de la fila de CTU.
7. Un medio de almacenamiento legible por ordenador que tiene instrucciones almacenadas en el mismo que, cuando se ejecutan, hacen que un procesador programable de un dispositivo de cálculo realice un procedimiento según una cualquiera de las reivindicaciones 1 a 3.

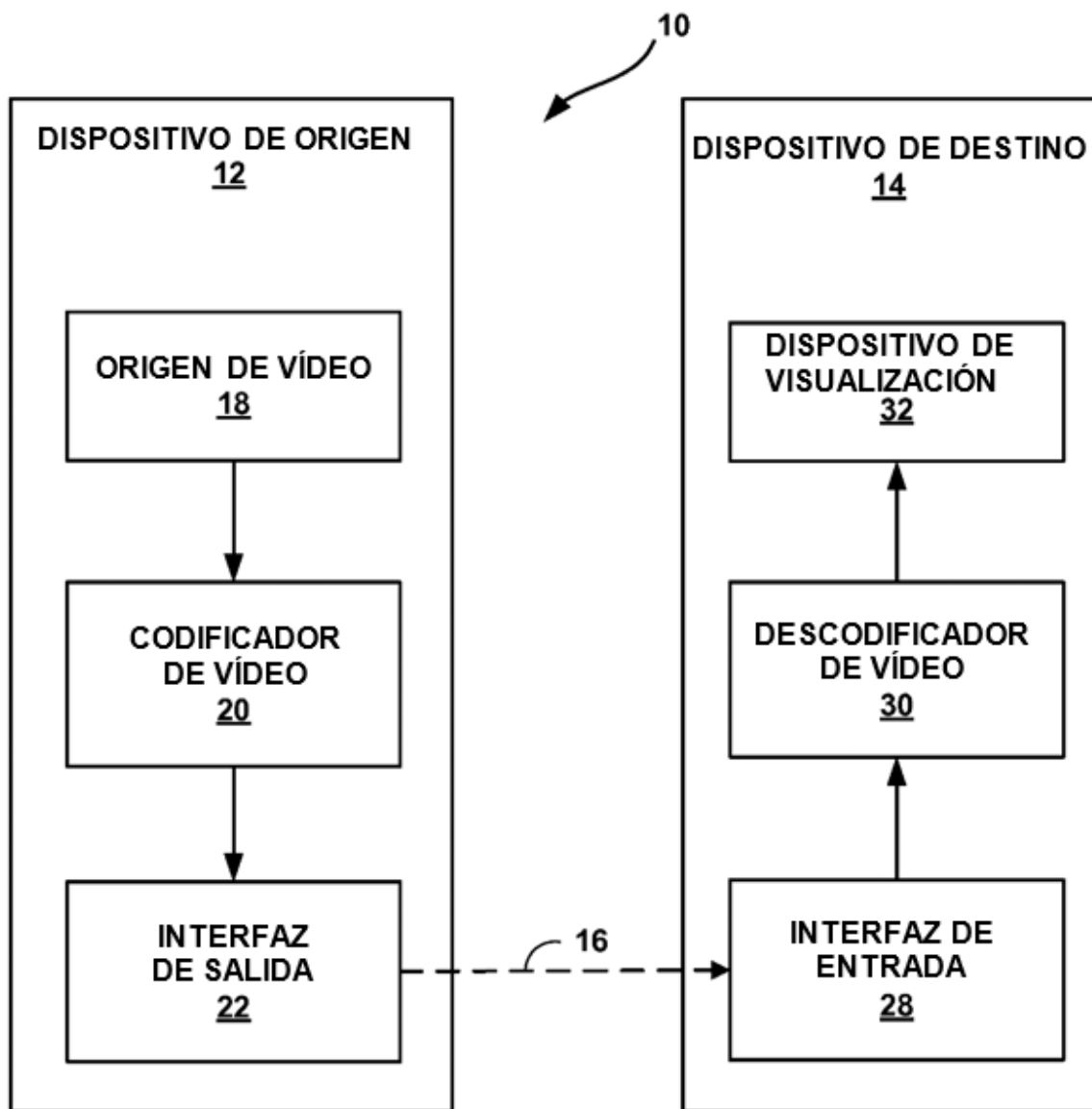


FIG. 1

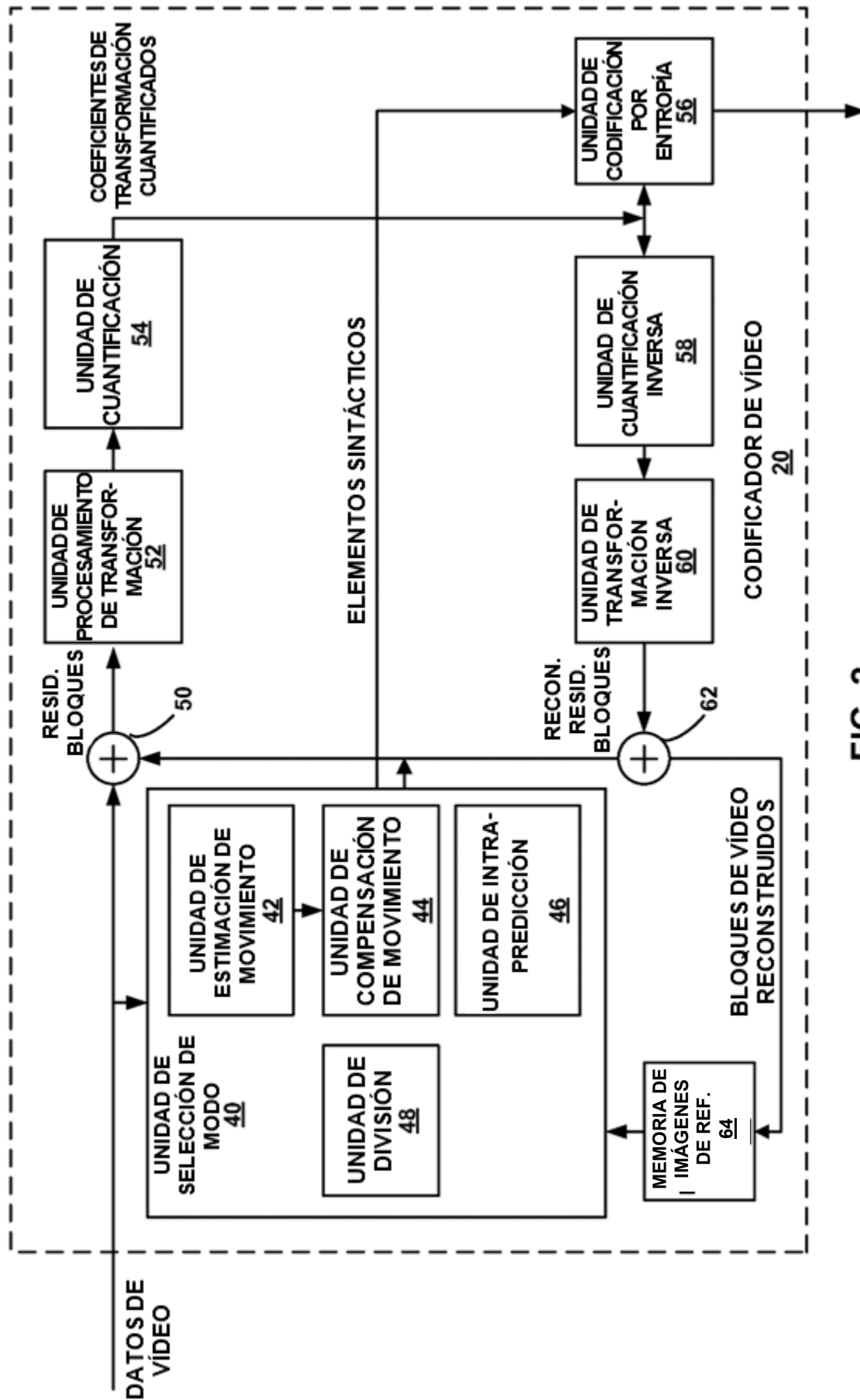


FIG. 2

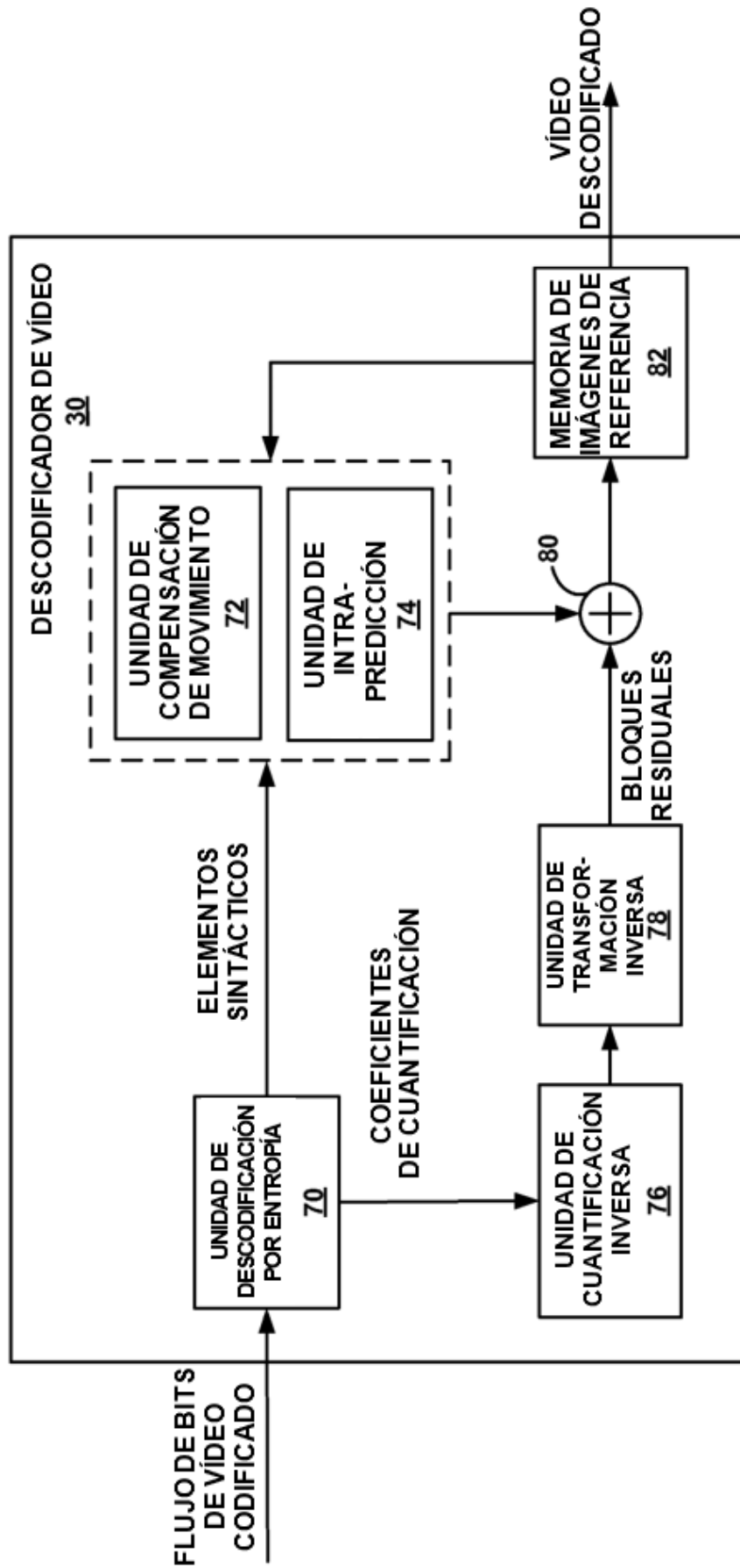


FIG. 3

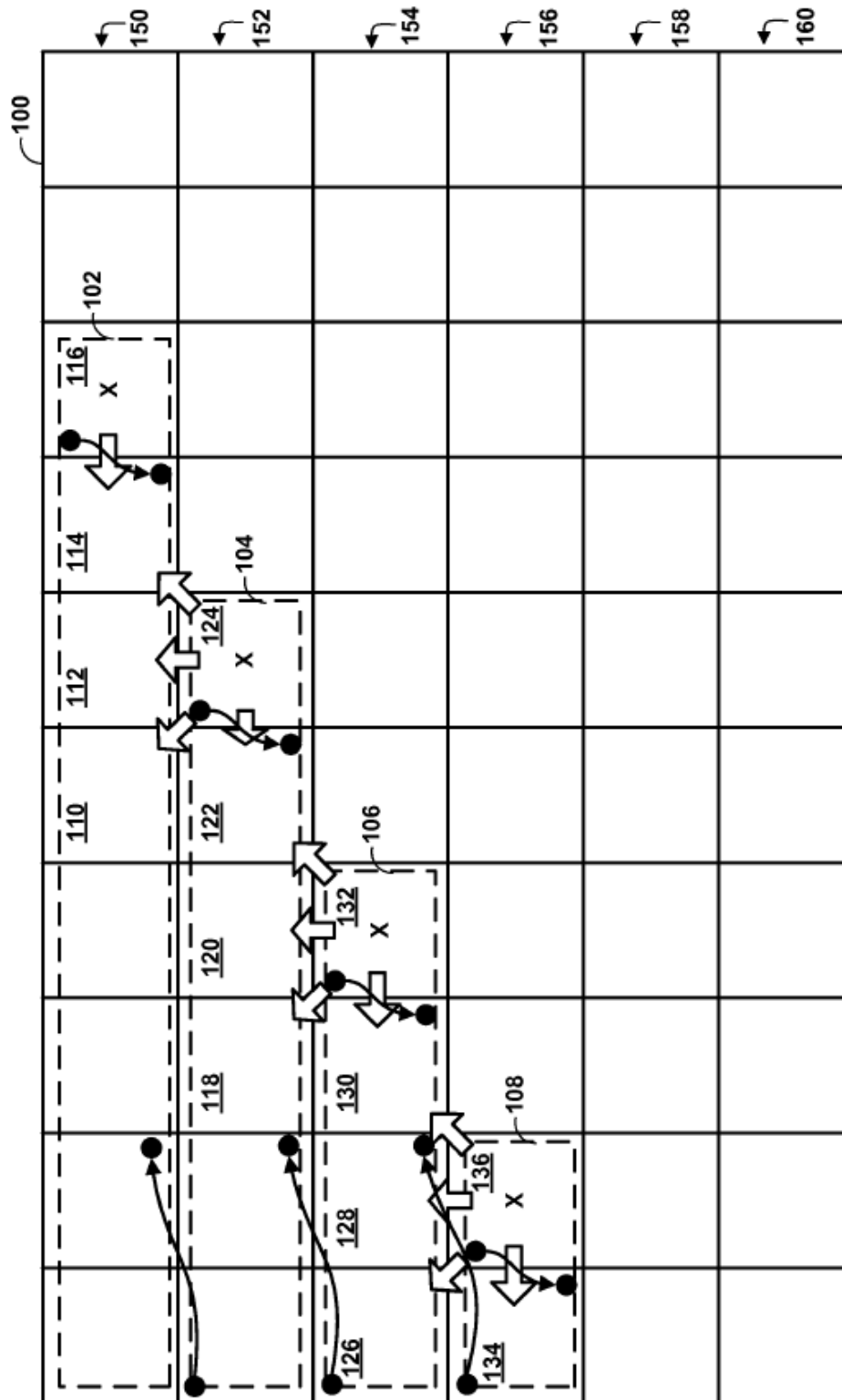


FIG. 4

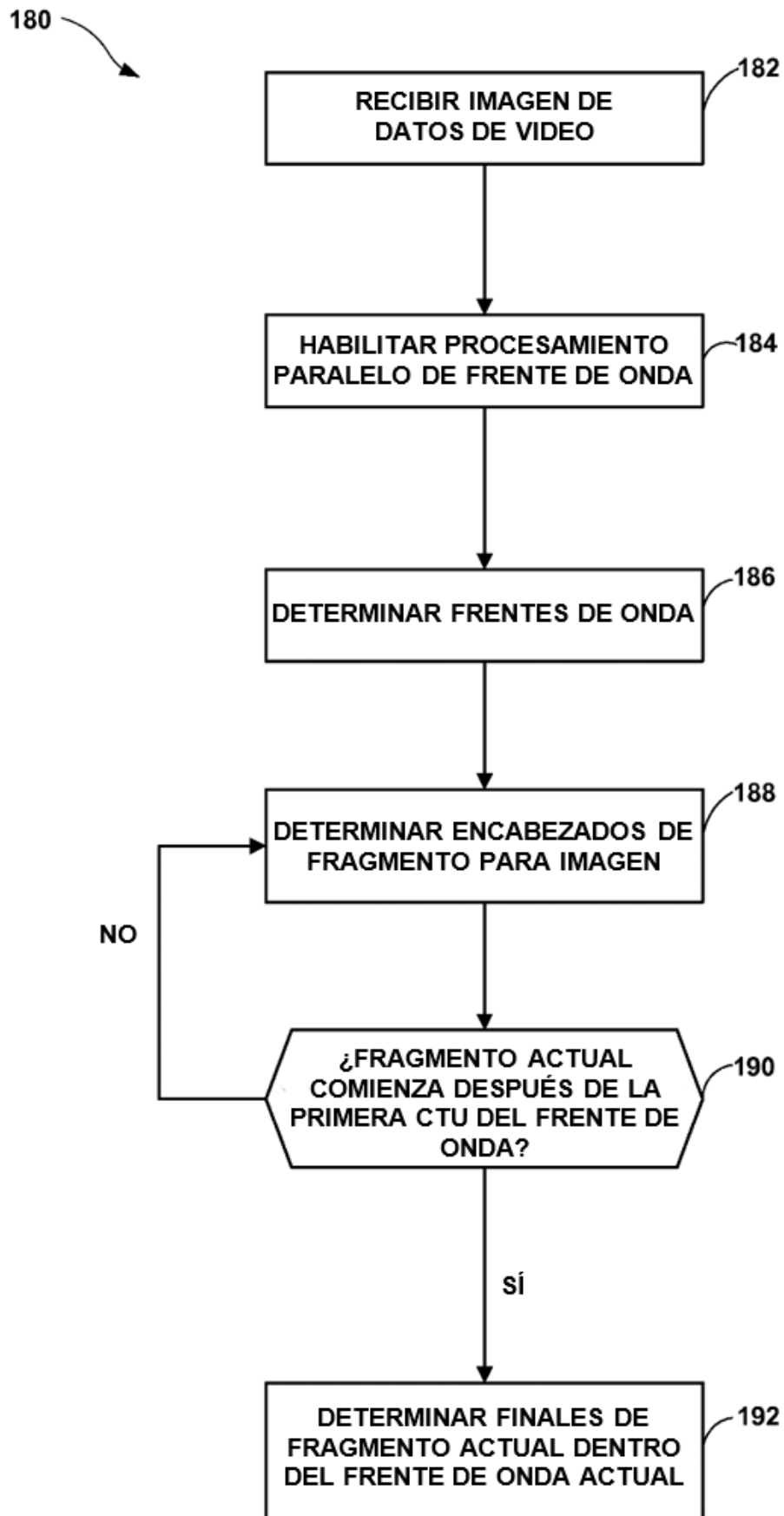


FIG. 5

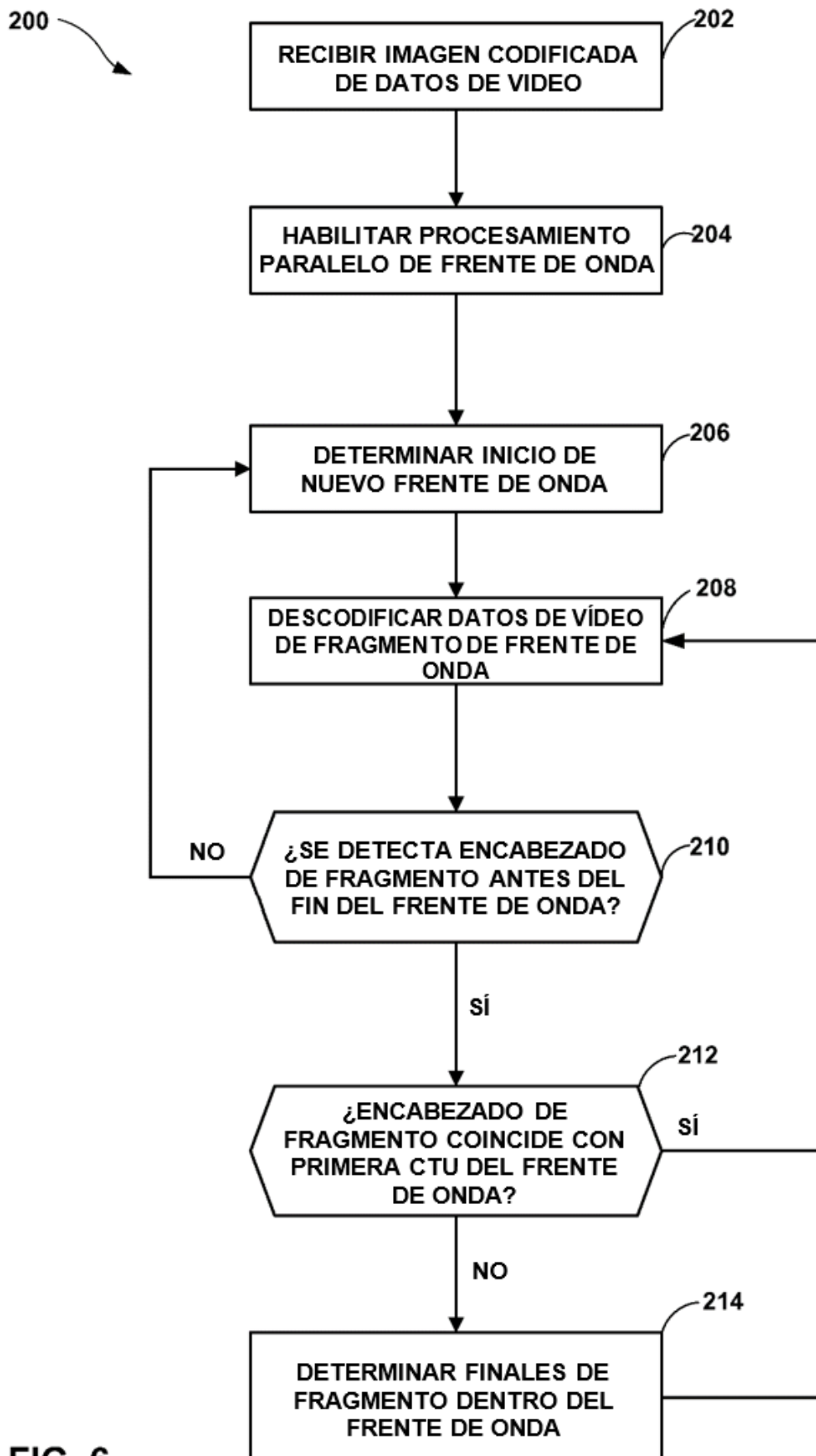


FIG. 6