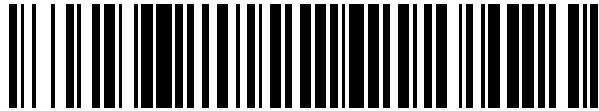


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 616 486**

51 Int. Cl.:

H04N 19/91 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **13.03.2013 PCT/US2013/030886**

87 Fecha y número de publicación internacional: **26.09.2013 WO2013142195**

96 Fecha de presentación y número de la solicitud europea: **13.03.2013 E 13712992 (0)**

97 Fecha y número de publicación de la concesión europea: **16.11.2016 EP 2829061**

54 Título: **Obtención de contexto para la codificación de última posición para la codificación de vídeo**

30 Prioridad:

22.03.2012 US 201261614178 P
04.04.2012 US 201261620273 P
29.06.2012 US 201261666316 P
12.03.2013 US 201313796572

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
13.06.2017

73 Titular/es:

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

72 Inventor/es:

GUO, LIWEI;
KARCZEWICZ, MARTA y
CHIEN, WEI-JUNG

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 616 486 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

Obtención de contexto para la codificación de última posición para la codificación de vídeo

5 CAMPO TÉCNICO

Esta divulgación se refiere a la codificación de vídeo.

ANTECEDENTES

10 Las capacidades del vídeo digital pueden incorporarse en una amplia gama de dispositivos, incluyendo televisores digitales, sistemas de difusión directa digital, sistemas de difusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, ordenadores de tableta, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos celulares o de radio por satélite, los denominados teléfonos "inteligentes", dispositivos de videoconferencia, dispositivos de transmisión por flujo 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, 15 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 extensiones de tales normas. 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.

25 Las técnicas de codificación de vídeo incluyen la predicción espacial (intra-imagen) y/o la predicción temporal (entre imágenes) 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 son codificados usando la predicción espacial con respecto a muestras de referencia en bloques contiguos en la misma imagen. Los bloques de vídeo de un fragmento inter-codificado (P o B) de una imagen pueden usar la predicción espacial con respecto a muestras de referencia en bloques contiguos en la misma imagen, o la 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.

35 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 según 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 según un modo de intra-codificación y los datos residuales. Para una mayor compresión, los datos residuales pueden transformarse desde el dominio de píxeles a un dominio de transformación, dando como resultado coeficientes de transformación residuales, los cuales pueden cuantificarse posteriormente. Los coeficientes de transformación cuantificados, inicialmente dispuestos en una formación bidimensional, pueden explorarse con el fin de producir un vector unidimensional de coeficientes de transformación, y puede aplicarse codificación por entropía para lograr aún más compresión.

45 La codificación del último coeficiente significativo se describe en el artículo "No-CE11: Procedimiento modificado para codificar las posiciones de los últimos coeficientes significativos en el modo CABAC "por S-THSIANG Y COL. publicado en 7. REUNIÓN DE JCTVC; 98 REUNIÓN MPEG; 21-11-2011 - 30-11-2011; GINEBRA; (EQUIPO DE COLABORACIÓN CONJUNTA SOBRE CODIFICACIÓN DE VÍDEO DE ISO/IEC JTC1/SC29/WG11 E ITU-T SG.16); URL; [HTTP://WFTP3.ITU.INT/AV-ARCH/JCTVC-SITE/](http://wftp3.itu.int/av-arch/jctvc-site/), nº JCTVC-G239, 8 de noviembre de 2011.

50 La codificación de última posición se describe en los artículos "Modificación de binarización para la codificación de última posición -JCTVC-F375_r2" de Vadim Segeren Y COL. publicada en la REUNIÓN DE JCTVC; 97. REUNIÓN MPEG; 14-7-2011 - 22-7-2011; TORINO; (EQUIPO DE COLABORACIÓN CONJUNTA SOBRE CODIFICACIÓN DE VÍDEO DE ISO/IEC JTC1/SC29/WG11 E ITU-T SG.16), 14 de julio de 2011, páginas 1-10, y "Reducción de Contextos CABAC para la Codificación de Última Posición - Revisión 3 (JCTVC-H0537r3)"por Liwei Guo y col. publicado el 3 febrero 2012, páginas 1-5.

SUMARIO

60 La invención se define en las reivindicaciones a las que se hace ahora referencia.

En general, esta divulgación describe técnicas para la codificación de elementos sintácticos asociados con datos de vídeo usando una o más funciones. Por ejemplo, un dispositivo puede implementar una o más de las técnicas para codificar un valor que indica una posición de un último coeficiente significativo de un bloque de datos de vídeo (tal como una unidad de transformación, o "TU"). Para codificar el valor, el dispositivo puede utilizar una función de un índice de cada bit (o "bin") en un valor binarizado correspondiente al último coeficiente significativo, donde el índice

indica una posición del bin en una serie de bins que representan el valor binarizado.

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

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 la determinación de un contexto a utilizar para codificar un valor que representa un último coeficiente significativo de un bloque de datos de vídeo.

La FIG. 2 es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo que puede implementar técnicas para la determinación de un contexto a utilizar para codificar un valor que representa un último coeficiente significativo de un bloque de datos de vídeo.

La FIG. 3 es un diagrama de bloques que ilustra un ejemplo de un descodificador de vídeo que puede implementar técnicas para la determinación de un contexto a utilizar para codificar un valor que representa un último coeficiente significativo de un bloque de datos de vídeo.

La FIG. 4 es un diagrama de flujo que ilustra un procedimiento de ejemplo para la codificación de un bloque actual de datos de vídeo.

La FIG. 5 es un diagrama de flujo que ilustra un procedimiento de ejemplo para descodificar un bloque actual de datos de vídeo.

DESCRIPCIÓN DETALLADA

En general, las técnicas de esta divulgación se refieren a la codificación de vídeo. En la codificación de vídeo, una secuencia de imágenes se codifican individualmente utilizando predicción espacial (intra-predicción) o predicción temporal (inter-predicción). En particular, los codificadores de vídeo codifican bloques individuales de las imágenes usando intra-predicción o inter-predicción. Los codificadores de vídeo también codifican datos residuales para los bloques, donde los datos residuales corresponden en general a bloques residuales, que representan diferencias de píxel por píxel entre los datos predichos y los datos no codificados, sin procesar. Los codificadores de vídeo pueden transformar y cuantificar los datos residuales para producir coeficientes de transformación cuantificados para los bloques residuales. Los codificadores de vídeo codifican además datos sintácticos, tales como si los coeficientes son significativos (por ejemplo, tienen valores absolutos mayores que cero), ubicaciones de los coeficientes significativos, una ubicación de un último coeficiente significativo en orden de exploración, y valores del nivel para los coeficientes significativos.

Esta divulgación describe técnicas para la codificación de un valor indicativo de un último coeficiente significativo en un bloque de datos de vídeo, tal como una unidad de transformación (TU). En particular, para codificar elementos sintácticos, tales como el valor indicativo del último coeficiente significativo en el bloque, los codificadores de vídeo pueden configurarse para aplicar una codificación aritmética binaria adaptable al contexto (CABAC). La codificación CABAC implica el uso de varios contextos, indicados por índices de contexto, que generalmente indican la probabilidad de que un bit (o "bin") individual de una cadena binarizada tenga un valor particular (por ejemplo, 0 o 1). Específicamente, el contexto para la codificación de un bin de un valor indicativo de un último coeficiente significativo en un bloque se determina individualmente para cada bin del valor, es decir, basándose en una ubicación del bin en el valor (por ejemplo, un índice del bin, suponiendo que el valor se representa como una serie de bins).

En lugar de utilizar una tabla de asignación, que proporciona indicaciones de los índices de contexto para los contextos a utilizar para codificar los bins particulares, las técnicas de esta divulgación incluyen el uso de una función para determinar el índice de contexto de un contexto a utilizar para codificar un bin. En particular, la función puede ser una función de un índice del bin. Por ejemplo, suponiendo que el bin es el *i*-ésimo bin de un valor que se está codificando, se puede definir una función como $f(i)$, donde $f(i)$ devuelve un valor de índice de contexto correspondiente a un contexto a utilizar para codificar el bin de un valor binarizado. El contexto, como se ha descrito anteriormente, puede indicar la probabilidad de que el bin *i* tenga un valor determinado, por ejemplo, 0 o 1.

De esta manera, esta divulgación describe técnicas de la codificación CABAC de la posición del último coeficiente significativo (última posición). Para la codificación de un bin de última posición, el índice de su contexto CABAC puede obtenerse utilizando una función, de forma que pueda guardarse una tabla de asignación entre los bins de última posición y los contextos CABAC (por ejemplo, no almacenados). La codificación CABAC generalmente incluye dos partes: la binarización y la codificación CABAC. El proceso de binarización se realiza para convertir la ubicación del último coeficiente significativo de un bloque en una secuencia binaria, por ejemplo, una serie de bins. El procedimiento de binarización usado en el modelo de prueba de codificación de vídeo de alta eficiencia (HM) es unario truncado + codificación de longitud fija. Para la parte de codificación unaria truncada, los bins se codifican

utilizando contextos CABAC. Para la parte de longitud fija, los bins se codifican utilizando el modo de derivación (sin contextos). Un ejemplo de TU 32x32 (unidad de transformación/bloque de transformación) se muestra en la tabla 1 a continuación:

5

TABLA 1

Magnitud del componente de la última posición	Unaria truncada (modelo de contexto)	Binaria fija (derivación)	f_valor
0	1	-	0
1	01	-	0
2	001	-	0
3	0001	-	0
4-5	00001	X	0-1
6-7	000001	X	0-1
8-11	0000001	XX	0-3
12-15	00000001	XX	0-3
16-23	000000001	XXX	0-7
24-31	000000000	XXX	0-7

La tabla 2 a continuación ilustra una tabla de asignación de contexto de ejemplo utilizada en el HM convencional. La tabla 2 muestra que las últimas posiciones en diferentes ubicaciones pueden compartir los mismos contextos. Para algunos bins, por ejemplo, los bins 6-7 de un bloque de 8x8, no hay contexto asignado; eso se debe a que están codificados sin contexto (modo de derivación), como se muestra en la tabla 1 anterior.

10

TABLA 2

Indice de bin	0	1	2	3	4	5	6	7	8	9
TU 4x4	0	1	2							
TU 8x8	3	4	5	5	2					
TU 16x16	6	7	8	8	9	9	2			
TU 32x32	10	11	12	14	13	13	14	14	2	

15

Aunque el HM convencional utiliza una tabla tal como la tabla 2 para determinar los contextos para la codificación de bins de un valor de la última posición (es decir, un valor que indica una posición del último coeficiente significativo en un bloque de datos de vídeo), las técnicas de esta divulgación incluyen el uso de una función para determinar los contextos para la codificación de bins del valor de la última posición. Por lo tanto, una tabla similar a la tabla 2 no tiene que estar necesariamente en un codificador de vídeo configurado según las técnicas de esta divulgación. De esta manera, se puede utilizar una función para obtener el índice de contexto de CABAC para la codificación de bins en la última posición, de forma que la tabla de asignación (tabla 2) se puede eliminar. A continuación se describen con mayor detalle varios ejemplos de dispositivos de codificación configurados para ejecutar funciones para determinar contextos para la codificación de bins de elementos sintácticos.

20

25

La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y decodificación de vídeo de ejemplo 10 que puede utilizar técnicas para la determinación de un contexto a utilizar para codificar un valor que representa un último coeficiente significativo de un bloque de datos de vídeo. 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 por parte de un dispositivo de destino 14. En particular, el dispositivo de origen 12 proporciona datos de vídeo al dispositivo de destino 14 a través de 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 portátiles 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 flujo 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.

30

35

El dispositivo de destino 14 puede recibir los datos de vídeo codificados que se van a descodificar, mediante un medio legible por ordenador 16. El medio legible por ordenador 16 puede comprender cualquier tipo de medio o

40

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 para permitir al dispositivo de origen 12 transmitir datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real. Los datos de vídeo codificados pueden ser modulados según una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitidos 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 extensa 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.

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, 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 ficheros 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 flujo o descarga. El servidor de ficheros puede ser cualquier tipo de servidor capaz de almacenar datos de vídeo codificados y transmitir esos datos de vídeo codificados al dispositivo de destino 14. Entre los ejemplos de servidores de ficheros se incluyen un servidor de Internet (por ejemplo, para un sitio web), un servidor de FTP, dispositivos de almacenamiento conectados a la red (NAS) o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de cualquier conexión de datos estándar, incluyendo una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión Wi-Fi), una conexión por cable (por ejemplo, DSL, módem de cable, etc.), o una combinación de ambos que sea adecuada para acceder a datos de vídeo codificados, almacenados en un servidor de ficheros. La transmisión de datos de vídeo codificados desde el dispositivo de almacenamiento puede ser una transmisión por flujo, una transmisión de descarga o una combinación de ambas.

Las técnicas de esta divulgación no están limitadas necesariamente a aplicaciones o configuraciones inalámbricas. Las técnicas pueden aplicarse a la codificación de vídeo, en soporte de cualquiera entre una diversidad de aplicaciones multimedia, tales como difusiones inalámbricas de televisión, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo por flujo mediante Internet, tal como el flujo adaptativo dinámico sobre HTTP (DASH), vídeo digital que está codificado en un medio de almacenamiento de datos, descodificación de vídeo digital almacenado en un medio de almacenamiento de datos, u otras aplicaciones. En algunos ejemplos, el sistema 10 puede configurarse para dar soporte a la transmisión de vídeo unidireccional o bidireccional, para prestar soporte a aplicaciones tales como el flujo de vídeo, la reproducción de vídeo, la difusión de vídeo y/o la videotelefonía.

En el ejemplo de la FIG. 1, el dispositivo de origen 12 incluye una fuente de vídeo 18, un codificador de vídeo 20 y una interfaz de salida 22. El dispositivo de destino 14 incluye una interfaz de entrada 28, un descodificador de vídeo 30 y un dispositivo de visualización 32. Según esta divulgación, el codificador de vídeo 20 de dispositivo de origen 12 puede estar configurado para aplicar las técnicas para la determinación de un contexto a utilizar para codificar un valor que representa un último coeficiente significativo de un bloque de datos de vídeo. 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.

El sistema ilustrado 10 de la figura 1 es simplemente un ejemplo. Las técnicas para determinar un contexto a utilizar para codificar un valor que representa un último coeficiente significativo de un bloque de datos de vídeo pueden realizarse mediante cualquier dispositivo de codificación y/o descodificación de vídeo digital. Aunque, por lo general, las técnicas de esta divulgación se llevan a cabo por 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 por un preprocesador de vídeo. El dispositivo de origen 12 y el dispositivo de destino 14 son simplemente ejemplos de tales dispositivos de codificación, donde 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 casi simétrica, de modo que cada uno de los dispositivos 12, 14 incluye componentes de codificación y de descodificación de vídeo. Por tanto, el sistema 10 puede soportar una transmisión de vídeo unidireccional o bidireccional entre los dispositivos de vídeo 12, 14, por ejemplo para el flujo de vídeo, la reproducción de vídeo, la radiodifusión de vídeo o la videotelefonía.

La fuente 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 contenido de vídeo. Como una alternativa adicional, la fuente de vídeo 18 puede generar vídeo fuente como datos basados en gráficos de ordenador, o una combinación de vídeo en directo, vídeo de un archivo y vídeo generado por ordenador. En algunos casos, si la fuente 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 aplicarse a la codificación de vídeo en general, y pueden aplicarse a aplicaciones inalámbricas y/o cableadas. En cada caso, el vídeo grabado, pregrabado o generado por ordenador puede codificarse por el codificador de vídeo 20. A continuación, la información de vídeo codificado puede ser emitida por la interfaz de salida 22 sobre un medio legible por ordenador 16.

El medio legible por ordenador 16 puede incluir medios transitorios, tales como una emisión inalámbrica o una transmisión por red cableada, o un medio de almacenamiento (es decir, medios de almacenamiento no transitorios), tal como un disco duro, una unidad 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, a través de una transmisión de red. Del mismo modo, un dispositivo informático de una instalación de producción de medios, 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, se puede entender que el medio legible por ordenador 16 incluye uno o más medios legibles por ordenador de varias formas, en varios ejemplos.

La interfaz de entrada 28 del dispositivo de destino 14 recibe información del medio legible por ordenador 16. La información del medio legible por ordenador 16 puede incluir información sintáctica definida por el codificador de vídeo 20, que también es usada por el descodificador de vídeo 30, que incluye elementos sintácticos que describen características y/o el 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 una variedad de dispositivos de visualización, tal como un tubo de rayos catódicos (CRT), una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos de emisión de luz (OLED) u otro tipo de dispositivo de visualización.

El codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar según una norma de compresión de vídeo, tal como la norma de Codificación de Vídeo de Alta Eficiencia (HEVC), actualmente en desarrollo, y pueden ser conformes al Modelo de Prueba de la HEVC (HM). Como alternativa, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar según otras normas patentadas o industriales, tales como la norma ITU-T H.264, también denominada MPEG -4, Parte 10, Codificación Avanzada de Vídeo (AVC), o extensiones de tales 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, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden estar integrados con un codificador y descodificador de audio, y pueden incluir unidades adecuadas de multiplexación y demultiplexación, u otro hardware y software, para llevar a cabo la codificación, tanto de audio como de vídeo, en un flujo de datos común o en flujos de datos diferentes. Si procede, las unidades de multiplexación y demultiplexación pueden ajustarse al protocolo de multiplexación ITU H.223 o a otros protocolos, tales como el protocolo de datagramas de usuario (UDP).

La norma ITU-T H.264/MPEG-4 (AVC) fue formulada por el Grupo de Expertos en Codificación de Vídeo (VCEG) de ITU-T junto con el Grupo de Expertos en Imágenes en Movimiento (MPEG) de ISO/IEC como el producto de una asociación colectiva conocida como el Equipo de vídeo Conjunto (NT). En algunos aspectos, las técnicas descritas en esta divulgación pueden aplicarse a dispositivos que generalmente cumplen con la norma H.264. La norma H.264 se describe en la Recomendación ITU-T H.264, Codificación de Vídeo Avanzada para los servicios audiovisuales genéricos, por el Grupo de Estudio de la ITU-T, y con fecha de marzo de 2005, que se puede denominar en este documento norma H.264 o memoria descriptiva H.264, o la norma o memoria descriptiva H.264/AVC. El Equipo de Vídeo Conjunto (NT) continúa trabajando en extensiones para H.264/MPEG-4 AVC.

El codificador de vídeo 20 y el descodificador de vídeo 30 pueden implementarse como cualquiera entre una diversidad de circuitos 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 campo (FPGA), lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. Cuando las técnicas son implementadas parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio no transitorio adecuado, legible por ordenador, y ejecutar las instrucciones en hardware usando uno o más procesadores para realizar las técnicas de esta divulgación. El codificador de vídeo 20 y el descodificador de vídeo 30 pueden incluirse en uno o más codificadores o descodificadores, donde cualquiera de los mismos puede estar integrado como parte de un codificador/descodificador combinado (CODEC) en un dispositivo respectivo.

El JCT-VC está trabajando en el desarrollo de la norma HEVC. Los esfuerzos de normalización de la HEVC se

basan en un modelo en evolución de un dispositivo de codificación de vídeo denominado modelo de prueba de la HEVC (HM). El HM supone varias capacidades adicionales de los dispositivos de codificación de vídeo, con respecto a dispositivos existentes, según, por ejemplo, la 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.

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 codificación más grandes (LCU), que incluyen muestras tanto de luma como de croma. Los datos sintácticos de un flujo de bits pueden definir un tamaño para la LCU, que es la mayor unidad de codificación en lo que respecta al número de píxeles. Un fragmento incluye un 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) según un árbol cuádruple. En general, una estructura de datos de árbol cuádruple incluye un nodo por CU, donde un nodo raíz corresponde al bloque arbolado. Si una CU se divide en cuatro sub-CU, el nodo correspondiente a la CU incluye cuatro nodos de hoja, cada uno de los cuales corresponde a una de las sub-CU.

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 de una CU pueden definirse de manera recursiva y pueden depender de si la CU está dividida o no 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 hoja 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.

Una CU tiene un propósito similar a un macro-bloque 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 nodo de hoja del árbol cuádruple, comprende un nodo de codificación, también denominado como 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 como profundidad de CU máxima, y también puede definir un tamaño mínimo de los nodos de codificación. Por consiguiente, un flujo de bits también puede definir una mínima unidad de codificación (SCU). Esta divulgación utiliza el término "bloque" para referirse a CU, PU, o TU, en el contexto de HEVC, o a estructuras de datos similares en el contexto de otras normas (por ejemplo, macrobloques y sub-bloques de los mismos en H.264/AVC).

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 8 x 8 píxeles hasta el tamaño del bloque arbolado, con un máximo de 64 x 64 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 entre si la CU está codificada en modo de omisión o directa, codificada en modo de intra-predicción o codificada en modo de inter-predicción. Las PU pueden dividirse para que no tengan forma cuadrada. Los datos sintácticos asociados a una CU también pueden describir, por ejemplo, la división de la CU en una o más TU según un árbol cuádruple. Una TU puede tener forma cuadrada o no cuadrada (por ejemplo, rectangular).

La norma HEVC admite transformaciones según las TU, que pueden ser distintas para distintas CU. Las TU son habitualmente dimensionadas en base al tamaño de las PU dentro de una CU dada, definida para una LCU dividida, aunque esto puede no ser siempre el caso. Las TU tienen habitualmente el mismo tamaño, o un tamaño más pequeño, que las PU. En algunos ejemplos, las muestras residuales correspondientes a una CU pueden subdividirse en unidades más pequeñas usando una estructura de árbol cuádruple conocida como "árbol cuádruple residual" (RQT). Los nodos de hoja del RQT pueden denominarse unidades de transformación (TU). Los valores de diferencias de píxeles asociados a las TU pueden transformarse para producir coeficientes de transformación, que pueden cuantificarse.

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 una porción 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, un componente horizontal del vector de movimiento, un componente vertical del vector de movimiento, una resolución para el vector de movimiento (por ejemplo, precisión de píxeles de un cuarto o precisión de píxeles de un octavo), 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.

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 una RQT (también denominada estructura de árbol cuádruple de TU), como la descrita 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 está dividida adicionalmente, puede denominarse como TU hoja. Generalmente, 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 generalmente 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 ubicada con una TU hoja correspondiente para la misma CU. En algunos ejemplos, el tamaño máximo de una TU hoja puede corresponder con el tamaño de la CU hoja correspondiente.

Además, las TU de las CU hoja también pueden asociarse a estructuras de datos de árbol cuádruple respectivas, denominadas como árboles cuádruples residuales (RQTs). 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 generalmente a una CU hoja, mientras que el nodo raíz de un árbol cuádruple de CU corresponde generalmente a un bloque arbolado (o LCU). Las TU del RQT que no están divididas se denominan TU hoja. 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.

Una secuencia de vídeo incluye normalmente una serie de tramas o imágenes de vídeo. Un grupo de imágenes (GOP) comprende generalmente una serie de una o más de las imágenes de vídeo. Un GOP puede incluir datos sintácticos, en una cabecera del GOP, en una cabecera 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 habitualmente sobre bloques de vídeo dentro de los fragmentos de vídeo individuales, con el fin de codificar los datos de vídeo. Un bloque de vídeo puede corresponder a un nodo de codificación dentro de una CU. Los bloques de vídeo pueden tener tamaños fijos o variables y pueden diferir en tamaño, según una norma de codificación especificada.

Como ejemplo, el HM da soporte a la predicción en diversos tamaños de PU. Suponiendo que el tamaño de una CU específica sea $2N \times 2N$, el HM presta soporte a la intra-predicción en tamaños de PU de $2N \times 2N$ o $N \times N$, y a la intra-predicción en tamaños de PU simétricos de $2N \times 2N$, $2N \times N$, $N \times 2N$ o $N \times N$. El HM también presta soporte a 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 entre el 25 % y el 75 %. La parte de la CU correspondiente a la partición del 25 % está indicada por una "n" seguida por una indicación de "Arriba", "Abajo", "Izquierda" o "Derecha". Así, por ejemplo, "2NxnU" se refiere a una CU de tamaño $2N \times 2N$ que está dividida horizontalmente, con una PU de tamaño $2N \times 0,5N$ encima y una PU de tamaño $2N \times 1,5N$ debajo.

En esta divulgación, "NxN" y "N por N" pueden usarse de manera intercambiable para hacer referencia a las dimensiones de píxeles de un bloque vídeo, en lo que respecta a 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 $N \times N$ tiene generalmente N píxeles en la dirección vertical y N píxeles en la dirección horizontal, donde N representa un valor entero no negativo. Los píxeles en un bloque pueden estar dispuestos en filas y columnas. Además, los bloques no necesitan tener 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.

Tras la codificación intra-predictiva o inter-predictiva, usando las PU de una CU, el codificador de vídeo 20 puede calcular datos residuales para las TU de la CU. Las PU pueden comprender datos sintácticos que describen un procedimiento o modo de generación de datos de píxeles predictivos en el dominio espacial (también denominado como 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 valores de predicción correspondientes a las PU. El codificador de vídeo 20 puede formar las TU incluyendo los datos residuales para la CU, y luego transformar las TU para producir coeficientes de transformación para la CU.

Tras cualquier transformación para producir coeficientes de transformación, el codificador de vídeo 20 puede realizar la cuantificación de los coeficientes de transformación. La cuantificación se refiere generalmente 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 una compresión adicional. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos de, o todos, los coeficientes. Por ejemplo, un valor de n bits puede redondearse por lo bajo a un valor de m bits durante la cuantificación, donde n es mayor que m .

5 Después de la cuantificación, el codificador de vídeo puede explorar los coeficientes de transformación, produciendo un vector unidimensional a partir de la matriz bidimensional, incluyendo los coeficientes de transformación cuantificados. La exploración puede estar diseñada para colocar los coeficientes de mayor energía (y por lo tanto de frecuencia más baja) en la parte delantera de la serie y colocar los coeficientes de menor energía (y por lo tanto de frecuencia más alta) en la parte posterior de la serie. En algunos ejemplos, el codificador de vídeo 20 puede utilizar un orden de exploración predefinido para explorar los coeficientes de transformación cuantificados, para producir un vector serializado 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 codificar por entropía el vector unidimensional, por ejemplo, según 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 de base sintáctica (SBAC), la codificación por entropía por división de intervalos de probabilidad (PIPE) u otra metodología de codificación por entropía. El codificador de vídeo 20 también puede codificar por entropía elementos sintácticos asociados a los datos de vídeo codificados, para su uso por el descodificador de vídeo 30, en la descodificación de los datos de vídeo.

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

30 Según las técnicas de esta divulgación, el codificador de vídeo 20 puede codificar un valor que representa una posición de un último coeficiente significativo de un bloque de datos de vídeo utilizando contextos determinados usando una o más funciones de bins del valor. Del mismo modo, el descodificador de vídeo 30 puede descodificar un valor que representa un último coeficiente significativo en un bloque de datos de vídeo utilizando contextos determinados usando una o más funciones de bins del valor. El codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden estar configurados para realizar cualquiera de las funciones (1)-(12), que se describen en mayor detalle a continuación, o funciones conceptualmente similares, para llevar a cabo las técnicas de esta divulgación. De esta manera, el codificador de vídeo 20 y el descodificador de vídeo 30 representan ejemplos de codificadores de vídeo configurados para determinar un contexto para la codificación por entropía de un bin de un valor indicativo de un último coeficiente significativo de un bloque de datos de vídeo usando una función de un índice del bin, y codificar el bin utilizando el contexto determinado.

A modo de ejemplo, "Ctx_i" puede denotar el índice de contexto utilizado por el codificador de vídeo 20 para codificar el i -ésimo bin en la cadena binaria de "última posición". El codificador de vídeo 20 puede obtener Ctx_i utilizando la siguiente ecuación:

$$45 \quad \text{Ctx}_i = f(i).$$

La función denotada por $f(i)$ puede ser lineal o no lineal. Además, $f(i)$ puede ser una función predefinida que es accesible tanto para el codificador de vídeo 20 como para el descodificador de vídeo 30. Como alternativa, $f(i)$ puede ser seleccionado por un usuario o por el codificador de vídeo 20, y transmitirse al descodificador de vídeo 30 utilizando uno o más tipos de señalización sintáctica de alto nivel, como un conjunto de parámetros de secuencia (SPS), un conjunto de parámetros de imagen (PPS), un conjunto de parámetros de adaptación (APS), una cabecera de trama, una cabecera de sección, una cabecera de secuencia, u otra señalización sintáctica de este tipo. Un ejemplo de una de tales funciones que el codificador de vídeo 20 puede ejecutar es:

$$55 \quad f(i) = (i \gg 1), \tag{1}$$

donde " \gg " denota el operador de desplazamiento a la derecha binario. A su vez, el resultado de $f(i)$ puede corresponder a Ctx_i. Es decir, el codificador de vídeo 20 puede ejecutar $f(i)$ para generar una salida igual al valor de Ctx_i. Más específicamente, el codificador de vídeo 20 puede ejecutar $f(i)$ para generar el índice de contexto de un contexto a utilizar para codificar por entropía el i -ésimo bin.

La tabla 3 siguiente ilustra un ejemplo de los índices de contexto que puede utilizar el codificador de vídeo 20 para codificar bins en diferentes índices de bin para varios tamaños de bloque (por ejemplo, TU) usando la función de ejemplo (1) descrita anteriormente. Aunque se proporciona la tabla 3 con el fin de explicar los resultados de la función de ejemplo (1), se apreciará que no es necesario que una tabla como la tabla 3 se almacene en un

dispositivo de codificación de vídeo, tal como el dispositivo de origen 12 y/o el dispositivo de destino 14. En su lugar, uno o ambos codificadores de vídeo 20 y el descodificador de vídeo 30 pueden ejecutar la función (1) anterior para producir los resultados indicados en la tabla 3, a partir de diversos índices de bin.

5

TABLA 3

Índice de bin	0	1	2	3	4	5	6	7	8	9
TU 4x4	0	0	1							
TU 8x8	0	0	1	1	2					
TU 16x16	0	0	1	1	2	2	3			
TU 32x32	0	0	1	1	2	2	3	3	4	

10 Como otro ejemplo, el codificador de vídeo 20 puede ejecutar una función que depende tanto del índice de bin (i) como de un tamaño de un bloque correspondiente (por ejemplo, una TU). El bloque correspondiente puede ser el bloque que incluye los coeficientes descritos por el valor del último coeficiente significativo. Como un ejemplo, el índice de contexto puede ser producido por una función, tal como:

15 $Ctx_i = f(i, TUBlkSize)$, donde "TUBlkSize" es un valor indicativo del tamaño de bloque. Para los fines de esta divulgación, los términos "TUBlkSize" y "block_size" se pueden usar indistintamente para indicar el tamaño de bloque.

Como un ejemplo, la función puede ser:

20 $f(i, TUBlkSize) = i \gg (\log_2(TUBlkSize) - 2)$. (2)

La tabla 4 siguiente ilustra un ejemplo de los índices de contexto que podría utilizar el codificador de vídeo 20 para codificar bins en diferentes índices de bin para varios tamaños de bloque (por ejemplo, TU) usando la función de ejemplo (2). Aunque la tabla 4 se proporciona con el fin de explicar los resultados de la función de ejemplo (2), se apreciará que no es necesario que una tabla como la tabla 4 se almacene en un dispositivo de codificación de vídeo, como el dispositivo de origen 12 y/o el dispositivo de destino 14. En su lugar, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden ejecutar la función de ejemplo (2) descrita anteriormente para producir los resultados indicados en la tabla 4.

30

TABLA 4

Índice de bin	0	1	2	3	4	5	6	7	8	9
TU 4x4	0	1	2							
TU 8x8	0	0	1	1	2					
TU 16x16	0	0	0	0	1	1	1			
TU 32x32	0	0	0	0	0	0	0	0	1	

35 Como otro ejemplo, el codificador de vídeo 20 puede ejecutar la siguiente función para obtener Ctx i:

$f(i, TUBlkSize) = i \gg 1 + TUSIZEoffset$, donde

$TUSIZEoffset = (\log_2(TUBlkSize) - 2) * (\log_2(TUBlkSize) + 1) / 2$. (3)

40 La tabla 5 siguiente ilustra un ejemplo de los índices de contexto que puede utilizar el codificador de vídeo 20 para codificar bins en diferentes índices de bin para varios tamaños de bloque (por ejemplo, TU) usando la función de ejemplo (3). Aunque la tabla 5 se proporciona con el fin de explicar los resultados de la función de ejemplo (3), se apreciará que no es necesario que una tabla como la tabla 5 se almacene en el dispositivo de origen 12 y/o en el dispositivo de destino 14. En su lugar, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden ejecutar la función de ejemplo (3) descrita anteriormente para producir los resultados indicados en la tabla 5.

45

50

TABLA 5

Indice de bin	0	1	2	3	4	5	6	7	8	9
TU 4x4	0	0	1							
TU 8x8	2	2	3	3	4					
TU 16x16	5	5	6	6	7	7	8			
TU 32x32	9	9	10	10	11	11	12	12	13	

Como otro ejemplo más, el codificador de vídeo 20 puede ejecutar la siguiente función para obtener Ctx_i:

5

$Ctx_idx = (i+1) \gg 1 + TUSIZEoffset$, donde

$TUSIZEoffset = (\log_2(TUBlkSize)-2) * (\log_2(TUBlkSize)+ 1)/2$.

10

Aunque la tabla 6 se proporciona con el fin de explicar los resultados de la función, se apreciará que no es necesario que una tabla como la tabla 6 se almacene en un dispositivo de codificación de vídeo, tal como el dispositivo de origen 12 y/o el dispositivo de destino 14. En su lugar, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden ejecutar la función de ejemplo (4) descrita anteriormente para producir los resultados indicados en la tabla 6.

15

TABLA 6

Indice de bin	0	1	2	3	4	5	6	7	8	9
TU 4x4	0	1	1							
TU 8x8	2	3	3	4	4					
TU 16x16	5	6	6	7	7	8	8			
TU 32x32	9	10	10	11	11	12	12	13	13	

20 Como otro ejemplo, la función puede ser:

$Ctx_idx = desplazamiento + (i \gg k)$, (5)

donde:

25

$desplazamiento = 3*n + ((n+1) \gg 2)$, (6)

$k = (n+3) \gg 2$, y (7)

30

$n = (\log_2(TUBlkSize)-2)$. (8)

Alternativamente, la función de ejemplo (8) se puede expresar como: $n = (\log_2(block_size)-2)$ para los fines de esta divulgación.

35

La tabla 7 siguiente ilustra un ejemplo de los índices de contexto que puede utilizar el codificador de vídeo 20 para codificar bins en diferentes índices de bin para varios tamaños de bloque (por ejemplo, TU) usando las funciones de ejemplo (5)-(8). Aunque la tabla 7 se proporciona con el fin de explicar los resultados de las funciones, se apreciará que no es necesario que una tabla como la tabla 7 se almacene en un dispositivo de codificación de vídeo, tal como el dispositivo de origen 12 y/o el dispositivo de destino 14. En su lugar, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden ejecutar las funciones de ejemplo (5)-(8) anteriores para producir los resultados indicados en la tabla 7.

45

TABLA 7

Indice de bin	0	1	2	3	4	5	6	7	8	9
TU 4x4	0	1	2							
TU 8x8	3	3	4	4	5					

TU 16x16	6	6	7	7	8	8	9			
TU 32x32	10	10	11	11	12	12	13	13	14	

Las tablas 8 y 9 siguientes ilustran otro ejemplo en el que el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden aplicar una o más técnicas de obtención de contexto basadas en fórmulas de esta divulgación para la codificación de bins en la "última posición" para componentes luma y croma de una manera unificada. En particular, la tabla 8 muestra los índices de bin para TU luma de diversos tamaños, mientras que la tabla 9 proporciona índices de bin para TU croma de diversos tamaños.

TABLA 8-Luma

Índice de bin	0	1	2	3	4	5	6	7	8	9
TU 4x4	0	1	2							
TU 8x8	3	3	4	4	5					
TU 16x16	6	6	7	7	8	8	9			
TU 32x32	10	10	11	11	12	12	13	13	14	

TABLA 9-Croma

Índice de bin	0	1	2	3	4	5	6	7	8	9
TU 4x4	0	1	2							
TU 8x8	0	0	1	1	2					
TU 16x16	0	0	1	1	2	2	3			

Un ejemplo de una función que pueden utilizar el codificador de vídeo 20 y/o el descodificador de vídeo 30 para obtener contextos para la codificación de bins en la última posición de TU luma, según la tabla 8, y de TU croma, según la tabla 9, es:

$$\text{Ctx_idx} = \text{desplazamiento} + (i \gg k), \tag{9}$$

donde luma y croma comparten el mismo valor de k, $k = (n+3) \gg 2$ con $n =$

$$(\log_2(\text{TUB1kSize}) - 2)$$

El codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden determinar los valores de la variable "desplazamiento" de la función (9), en función de si la TU es una TU luma o una TU croma utilizando diferentes funciones. Ejemplos de tales funciones incluyen las siguientes:

$$\text{Luma: desplazamiento} = 3 * n + ((n + 1) \gg 2) \tag{10}$$

$$\text{Croma: desplazamiento} = 0 \tag{11}$$

De esta manera, la función (9) representa un ejemplo de una función que pueden ejecutar el codificador de vídeo 20 y/o el descodificador de vídeo 30 para producir un índice de contexto. A su vez, el índice de contexto puede ser indicativo de un contexto para la codificación de un bin de un valor indicativo de un último coeficiente significativo de un bloque de datos de vídeo en función de un índice del bin (i) y un valor indicativo de un tamaño del bloque (k, que se calcula en función de n, que es $\log_2(\text{TUB1kSize}) - 2$). En este ejemplo, el codificador de vídeo 20 y/o el descodificador de vídeo 30 también pueden ejecutar la función ejemplo (9) para producir el índice de contexto basándose en un valor de desplazamiento que se determina en función de si el bloque es un bloque croma o un bloque luma, por ejemplo, como se muestra en las funciones (10) y (11).

Como otro ejemplo, el codificador de vídeo 20 puede implementar una función de paso para obtener el índice de contexto a utilizar para codificar por entropía el *i*-ésimo bin. Más específicamente, la función de paso puede representar una función que tiene dos o más partes dependiendo de, por ejemplo, el valor del índice de bin *i*. Por lo tanto, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden dividir los bins en el valor de la última posición en diferentes subconjuntos, por ejemplo; subconjunto0, subconjunto1, etc.

Además, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden aplicar diferentes funciones para diferentes subconjuntos, por ejemplo, F0() para subconjunto0, F1() para subconjunto1, y así sucesivamente. Por ejemplo, una función de este tipo puede ser de la forma siguiente:

$$f(i, TUBlkSize) = \begin{cases} i = last_bin, & 10 \\ else & (i \gg 1) + TUSIZEOffset \end{cases}, \text{ donde}$$

$$TUSIZEoffset = (\log_2(TUBlkSize)-2) * (\log_2(TUBlkSize)-1)/2. \quad (12)$$

En algunas implementaciones, los subconjuntos pueden estar definidos previamente, y la definición de los subconjuntos puede ser accesible tanto para el codificador de vídeo 20 como para el descodificador de vídeo 30.

Alternativamente, el codificador de vídeo 20 (o un usuario del dispositivo de origen 12) puede seleccionar los subconjuntos, y la interfaz de salida 22 puede transmitir los subconjuntos seleccionados al descodificador de vídeo 30 del dispositivo de destino 14 utilizando una o más técnicas de señalización sintáctica de alto nivel, tales como un SPS, un PPS, un APS, una cabecera de trama, una cabecera de fragmento, una cabecera de secuencia, u otra señalización sintáctica de este tipo. La definición de los subconjuntos también puede depender de diversos tipos de información diferentes, tales como el tamaño de bloque (por ejemplo, el tamaño de la TU), la profundidad del árbol cuádruple residual (RQT), la profundidad correspondiente al bloque, si el bloque corresponde a un componente de luminancia o un componente de crominancia, el tamaño de trama para la trama que incluye el bloque (por ejemplo, en resolución de píxeles), el tamaño de bloque de compensación de movimiento para un bloque de compensación de movimiento (por ejemplo, una unidad de predicción (PU)) correspondiente al bloque, el tipo de trama (I/P/B) para la trama que incluye el bloque, la dirección de inter-predicción para el bloque de compensación de movimiento correspondiente, la amplitud del vector de movimiento para el bloque de compensación de movimiento correspondiente, y/o la amplitud de una diferencia de vectores de movimiento para el vector de movimiento del bloque de compensación de movimiento correspondiente.

La tabla 10 siguiente ilustra un ejemplo de los índices de contexto que puede utilizar el codificador de vídeo 20 para codificar bins en diferentes índices de bin para varios tamaños de bloque (por ejemplo, TU) usando la función de ejemplo (12). Aunque la tabla 10 se proporciona con el fin de explicar los resultados de la función, se apreciará que no es necesario que una tabla tal como la tabla 10 esté almacenada en un dispositivo de codificación de vídeo, tal como el dispositivo de origen 12 y/o el dispositivo de destino 14. En su lugar, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden ejecutar la función de ejemplo (12) descrita anteriormente para producir los resultados indicados en la tabla 10.

TABLA 10

Índice de bin	0	1	2	3	4	5	6	7	8	9
TU 4x4	0	0	10							
TU 8x8	1	1	2	2	10					
TU 16x16	3	3	4	4	5	5	10			
TU 32x32	6	6	7	7	8	8	9	9	10	

Las funciones de ejemplo (1) - (12) descritas anteriormente pueden depender, al menos en parte, de uno o más elementos de información lateral. Como ejemplo, las funciones pueden aceptar la información lateral como argumentos. En otros ejemplos, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden seleccionar diferentes funciones en base a la información lateral correspondiente.

La información lateral puede incluir cualquiera o todos los tamaños de bloque (por ejemplo, el tamaño de la TU), la profundidad de árbol cuádruple residual (RQT), la profundidad correspondiente al bloque, si el bloque corresponde a un componente de luminancia o a un componente de crominancia, el tamaño de trama para la trama que incluye el bloque (por ejemplo, en resolución de píxeles), el tamaño de bloque de compensación de movimiento para un bloque de compensación de movimiento (por ejemplo, una unidad de predicción (PU)) correspondiente al bloque, el tipo de trama (I/P/B) para la trama que incluye el bloque, la dirección de inter-predicción para el bloque de compensación de movimiento correspondiente, la amplitud del vector de movimiento para el bloque de compensación de movimiento correspondiente, y/o una amplitud de la diferencia de vectores de movimiento para el vector de movimiento del bloque de compensación de movimiento correspondiente. Como ejemplo, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden seleccionar diferentes funciones para obtener contextos para aplicar al codificar los bins de un valor que indica una posición del último componente significativo de un bloque de luminancia, con respecto a un bloque de crominancia.

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 una cabecera de trama, una cabecera de bloque, una cabecera de fragmento o una cabecera de GOP. Los datos sintácticos de GOP pueden describir un cierto número de tramas en el GOP respectivo, y los datos sintácticos de trama pueden indicar un modo de codificación/predicción usado para codificar la trama correspondiente.

El codificador de vídeo 20 y el descodificador de vídeo 30 pueden implementarse como cualquiera entre una amplia variedad de sistemas 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), circuito de lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. Cada uno, entre el codificador de vídeo 20 y el descodificador de vídeo 30, puede estar incluido en uno o más codificadores o descodificadores, cada uno de los cuales puede estar integrado como parte de un codificador/descodificador (CODEC) de vídeo combinado. Un dispositivo que incluye el codificador de vídeo 20 y/o el descodificador de vídeo 30 puede comprender un circuito integrado, un microprocesador y/o un dispositivo de comunicación inalámbrica, tal como un teléfono celular.

De esta manera, el codificador de vídeo 20 y el descodificador de vídeo 30 representan ejemplos de un codificador de vídeo configurado para determinar un contexto para la codificación por entropía de un bin de un valor indicativo de un último coeficiente significativo de un bloque de datos de vídeo usando una función de un índice del bin, y codificar el bin utilizando el contexto determinado.

La FIG. 2 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo 20 que puede implementar técnicas para la determinación de un contexto a utilizar para codificar un valor que representa un último coeficiente significativo de un bloque de datos de vídeo. 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 apoya 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 apoya 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.

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 tramas 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 una unidad de cuantificación inversa 58, una unidad de transformación inversa 60 y un sumador 62. También puede incluirse un filtro de desbloqueo (no mostrado 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 filtraría habitualmente la salida del sumador 62. También pueden usarse filtros adicionales (en bucle o pos-bucle), además del filtro de desbloqueo. Estos 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).

Durante el proceso de codificación, el codificador de vídeo 20 recibe una trama o un fragmento de vídeo que va a codificarse. La trama o el fragmento puede estar dividido 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, como alternativa, puede llevar a cabo la codificación intra-predictiva del bloque de vídeo recibido, con respecto a uno o más bloques contiguos en 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 adecuada para cada bloque de datos de vídeo.

Además, la unidad de división 48 puede dividir bloques de datos de vídeo en sub-bloques, en base a 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 LCU, y dividir cada una de las LCU en sub-CU, basándose en un análisis de distorsión de velocidad (por ejemplo, optimización de distorsión de velocidad). La unidad de selección de modo 40 puede producir además una estructura de datos de árbol cuádruple que indica la división de una LCU en varias sub-CU. Las CU de nodos de hojas del árbol cuádruple pueden incluir una o más PU y una o más TU.

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 error, 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 usarse 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.

La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden estar sumamente integradas, pero se ilustran por separado con fines conceptuales. La estimación de movimiento, realizada por la unidad de estimación de movimiento 42, es el proceso de generación de vectores de movimiento, que estiman el movimiento para los bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de una PU de un bloque de vídeo dentro de una trama o imagen actual de vídeo, con respecto a un bloque predictivo dentro de una trama de referencia (u otra unidad codificada), con respecto al bloque actual que está siendo codificado 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, 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 tramas de referencia 64. Por ejemplo, el codificador de vídeo 20 puede interpolar valores de posiciones de un cuarto de píxel, posiciones de un octavo de píxel u otras posiciones fraccionarias de píxel de la imagen de referencia.

Por lo tanto, la unidad de estimación de movimiento 42 puede realizar una búsqueda de movimiento con respecto a las posiciones de píxeles completos y a las posiciones de fracciones de píxel, y emitir un vector de movimiento con una precisión de fracciones de píxel.

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 a partir 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 tramas 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.

La compensación de movimiento, llevada a cabo por la unidad de compensación de movimiento 44, puede implicar capturar o generar el bloque predictivo en base al 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 está siendo codificado, generando valores de diferencia de píxel, como se expone posteriormente. En general, la unidad de estimación de movimiento 42 lleva a cabo una estimación de movimiento con respecto a las componentes de luminancia, y la unidad de compensación de movimiento 44 usa vectores de movimiento calculados en función de las componentes de luminancia para las componentes de crominancia y las componentes de luminancia. La unidad de selección de modo 40 también puede generar elementos sintácticos asociados a los bloques de vídeo y al fragmento de vídeo para su uso por parte del descodificador de vídeo 30 a la hora de descodificar los bloques de vídeo del fragmento de vídeo.

La unidad de intra-predicción 46 puede intra-predecir 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 usar 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 adecuado de intra-predicción para usar entre los modos probados.

Por ejemplo, la unidad de intra-predicción 46 puede calcular valores de distorsión de velocidad usando un análisis de distorsión de velocidad para los diversos modos de intra-predicción probados, y seleccionar el modo de intra-predicción que tenga las mejores características de distorsión de velocidad entre los modos probados. El análisis de distorsión de velocidad determina generalmente una magnitud de distorsión (o 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, el número de bits) usada para generar el bloque codificado. La unidad de intra-predicción 46 puede calcular relaciones a partir de las distorsiones y velocidades para los diversos bloques codificados, para determinar qué modo de intra-predicción presenta el mejor valor de distorsión de velocidad para el bloque.

Después de seleccionar un modo de intra-predicción para un bloque, la unidad de intra-predicción 46 puede proporcionar información, que indica el 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 que indica 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 modificados (también denominadas tablas de correlación 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 modificada de índices de modos de intra-predicción, a usar en cada uno de los contextos.

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 está siendo codificado. El sumador 50 representa el componente o los componentes que realizan esta operación de resta. La unidad de procesamiento de transformaciones 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 transformaciones 52 puede llevar a cabo otras transformaciones que son conceptualmente similares a la DCT. También podrían usarse transformaciones de ondículas, transformaciones de enteros, transformaciones de sub-bandas u otros tipos de transformaciones. En cualquier caso, la unidad de procesamiento de transformaciones 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 transformaciones 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 de bits. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos de, o a todos, 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, incluyendo los coeficientes de transformación cuantificados. Como alternativa, la unidad de codificación por entropía 56 puede realizar la exploración.

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 guardarse para su posterior transmisión o recuperación.

En particular, la unidad de codificación por entropía 56 puede recibir, de la unidad de cuantificación 54, un conjunto de coeficientes de transformación cuantificados asociados a una TU. A su vez, la unidad de codificación por entropía 56 puede explorar el conjunto de coeficientes de transformación cuantificados, y determinar si cada coeficiente explorado incluye un coeficiente significativo, es decir, si el coeficiente tiene un valor de cero o distinto de cero. Un valor distinto de cero puede indicar que un determinado coeficiente de transformación cuantificado es un coeficiente "significativo". En los casos en que la unidad de codificación por entropía 56 detecta un coeficiente significativo, la unidad de codificación por entropía 56 puede codificar datos representativos del valor particular asociado con el coeficiente (por ejemplo, uno, dos, etc.). Tales datos pueden incluir, por ejemplo, una indicación del signo del coeficiente, si el valor absoluto del coeficiente es mayor que uno, y cuando el valor absoluto del coeficiente es mayor que uno, si el valor absoluto del coeficiente es mayor que dos. Además, en los casos en que un coeficiente significativo tiene un valor absoluto mayor que dos, la unidad de codificación por entropía 56 puede restar dos al valor absoluto del coeficiente, obteniendo de este modo un valor por el que el coeficiente es superior a dos, y codificar este valor.

Mediante la exploración de todo el conjunto de coeficientes de transformación cuantificados recibidos de la unidad de cuantificación 54, la unidad de codificación por entropía 56 también puede detectar e identificar el último coeficiente significativo asociado con una TU particular (es decir, en el orden de exploración). Además, la unidad de codificación por entropía 56 puede determinar la posición del último coeficiente significativo en la TU correspondiente. Por ejemplo, la unidad de codificación por entropía 56 puede identificar las coordenadas horizontal y vertical (x- e y-) del último coeficiente significativo dentro de la TU.

Por otra parte, la unidad de codificación por entropía 56 puede estar configurada para binarizar elementos sintácticos que todavía no tienen un valor binario. Es decir, la unidad de codificación por entropía 56 puede determinar una cadena binaria representativa del valor de un elemento sintáctico cuando el elemento sintáctico no está ya representado por una secuencia binaria. Una cadena binaria, o valor binarizado, generalmente corresponde a una serie de bits, cada uno de los cuales puede tener un valor de "0" o "1". La matriz puede estar indexada a cero, de tal manera que el primer bit ordinal de la serie se produce en posición 0, el segundo bit ordinal de la serie se produce en la posición 1, y así sucesivamente. Por lo tanto, la unidad de codificación por entropía 56 puede formar un valor binarizado $B[N]$ que tiene una longitud de N bits, donde cada bit se produce en una respectiva posición $B[i]$, donde 0

≤ i ≤ N-1.

A su vez, la unidad de codificación por entropía 56 puede codificar por entropía los datos que representan las coordenadas x- e y- del último coeficiente significativo. Por ejemplo, la unidad de codificación por entropía 56 puede estar configurada para codificar por entropía los elementos sintácticos prefijo_x_de_último_componente_significativo, prefijo_y_de_último_componente_significativo, sufijo_x_de_último_componente_significativo y/o sufijo_y_de_último_componente_significativo, que en conjunto, en HEVC, representan las coordenadas x- e y- del último coeficiente significativo en orden de exploración. La unidad de codificación por entropía 56 puede implementar una o más técnicas de esta divulgación para codificar por entropía datos que representan las coordenadas del último coeficiente significativo utilizando una función, que se denota por f(i). Por ejemplo, la unidad de codificación por entropía 56 puede codificar por entropía diversos elementos sintácticos, tales como los elementos sintácticos para los coeficientes de transformación cuantificados recibidos de la unidad de cuantificación 54 y/o los valores que representan un último coeficiente significativo de una TU (por ejemplo, los elementos sintácticos descritos anteriormente), utilizando contextos determinados usando una o más funciones de bins de un valor representativo del elemento sintáctico correspondiente.

Por ejemplo, "Ctx_i" puede denotar el índice de contexto utilizado por la unidad de codificación por entropía 56 para codificar el i-ésimo bin en un valor binarizado que representa la posición del último coeficiente significativo, como se ha descrito anteriormente con respecto a las tablas 1-2 y 8-9. El contexto indexado por ctx_i generalmente indica un símbolo más probable (por ejemplo, "1" o "0"), así como la probabilidad del símbolo más probable. La unidad de codificación por entropía 56 puede obtener el valor de Ctx_i usando la ecuación $Ctx_i = f(i)$, donde f(i) puede ser una función predefinida accesible a la unidad de codificación por entropía 56, o una función seleccionada por un usuario.

Además, la unidad de codificación por entropía 56 puede codificar datos representativos de f(i), de modo que el decodificador de vídeo 30 puede decodificar los datos para la función f(i) y usar f(i) para obtener el valor de Ctx_i. De esta manera, la unidad de codificación por entropía 56 puede determinar el contexto para un bin particular de un elemento sintáctico binarizado utilizando una función del índice de bin, es decir, la posición del bin en un valor binarizado (es decir, una cadena binaria) que representa el elemento sintáctico.

En algunos ejemplos, la unidad de codificación por entropía 56 se configura para determinar los contextos para codificación de bins de datos que representan la posición del último coeficiente significativo utilizando las fórmulas (5)-(8) descritas anteriormente. Es decir, la unidad de codificación por entropía 56 puede calcular f(i) de la forma siguiente: $Ctx_idx = desplazamiento + (i >> k)$. Por otra parte, la unidad de codificación por entropía 56 puede obtener los valores del valor de desplazamiento y k utilizados en f(i) usando las siguientes ecuaciones:

$$desplazamiento == 3 * n + ((n + 1) >> 2),$$

$$k = (n + 3) >> 2, \text{ y}$$

$$n = (\log_2(\text{block_size}) - 2).$$

En otras implementaciones, la unidad de codificación por entropía 56 puede utilizar una o más funciones de ejemplo (1)-(4) y (9)-(12), además o como alternativa a las fórmulas (5)-(8), al determinar un contexto para la codificación por entropía de un bin de datos que representa la posición del último coeficiente significativo de una TU. De esta manera, el codificador de vídeo 20 y los componentes del mismo, tales como la unidad de codificación por entropía 56, pueden implementar las técnicas de esta divulgación para codificar datos representativos del último coeficiente significativo usando una o más funciones. Dichas funciones se pueden almacenar de manera más eficiente en la memoria del codificador de vídeo 20 y el decodificador de vídeo 30 que en tablas. Por lo tanto, las técnicas de esta divulgación pueden proporcionar codificadores y decodificadores de vídeo de vídeo que utilizan la memoria más eficientemente, por ejemplo, mediante la asignación de memoria que de otro modo se dedicaría a una tabla para otros datos, o mediante la reducción de la cantidad necesaria de memoria para un codificador de vídeo o un decodificador de vídeo.

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 tramas 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 fracciones de píxel 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 tramas de referencia 64. El bloque de vídeo reconstruido puede ser usado 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 posterior.

De esta manera, el codificador de vídeo 20 de la FIG. 2 representa un ejemplo de un codificador de vídeo

configurado para determinar un contexto para la codificación por entropía un bin de un valor indicativo de un último coeficiente significativo de un bloque de datos de vídeo usando una función de un índice del bin, y codificar el bin usando el contexto determinado. Por otra parte, el codificador de vídeo 20 también representa un ejemplo de un codificador de vídeo en el que la función produce un índice de contexto para el contexto desplazando a la derecha el índice del bin un valor k y añadiendo el valor desplazado a la derecha a un valor de desplazamiento, en donde el valor de desplazamiento se determina según la fórmula desplazamiento = $3*n + ((n+1) \gg 2)$, donde el valor k se determina según la fórmula $k = (n+3) \gg 2$, y donde el valor n se determina según la fórmula $n = (\log_2(\text{block_size}) - 2)$.

La FIG. 3 es un diagrama de bloques que ilustra un ejemplo de un descodificador de vídeo 30 que puede implementar técnicas para la determinación de un contexto a utilizar para codificar un valor que representa un último coeficiente significativo en un bloque de datos de vídeo. 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 tramas 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 basados 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 basados en indicadores de modo de intra-predicción recibidos desde la unidad de descodificación por entropía 70.

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 al nivel del fragmento de vídeo y/o al nivel del bloque de vídeo.

La unidad de descodificación por entropía 70 puede generar un bloque (por ejemplo, una TU) de coeficientes cuantificados mediante la descodificación por entropía del flujo de bits de vídeo codificado, y llenar los coeficientes cuantificados descodificados por entropía en el bloque en orden de exploración. Por ejemplo, la unidad de descodificación por entropía 70 puede descodificar por entropía elementos sintácticos del flujo de bits de vídeo codificado para determinar las ubicaciones de los coeficientes significativos en el bloque a generar. Si una ubicación del bloque corresponde a un coeficiente que no es un coeficiente significativo, la unidad de descodificación por entropía 70 puede establecer el valor del coeficiente en esa ubicación en el bloque a cero. Por otro lado, si la unidad de descodificación por entropía determina que un coeficiente cuantificado en particular es un coeficiente significativo, la unidad de descodificación por entropía 70 puede establecer el valor del coeficiente significativo en base a los datos proporcionados en el flujo de bits de vídeo codificado por el codificador de vídeo 20.

Además, como se explica a continuación, la unidad de descodificación por entropía 70 puede determinar la posición de un último coeficiente significativo en el bloque en base a los elementos sintácticos que indican las coordenadas x - e y - del último coeficiente significativo. Según las técnicas de esta divulgación, como se explica en mayor detalle a continuación, la unidad de descodificación por entropía 70 puede utilizar una función para determinar el contexto para la descodificación por entropía de bins de valores que representan las coordenadas x - e y - del último coeficiente significativo. El descodificador de vídeo 30 puede utilizar la indicación de la posición del último coeficiente significativo para determinar cuándo los datos del flujo de bits representan elementos sintácticos posteriores, es decir, elementos sintácticos que no representan datos del bloque que se está regenerando.

La unidad de descodificación por entropía 70 puede determinar, en base a los datos proporcionados en el flujo de bits de vídeo codificado, un signo para cada coeficiente significativo, y los datos que representan el valor del nivel de cada coeficiente significativo. Por ejemplo, la unidad de descodificación por entropía 70 puede determinar un signo para un coeficiente significativo a través de descodificación por entropía de un elemento sintáctico que representa el signo, por ejemplo, indicador_signo_coeficiente. Además, la unidad de descodificación por entropía 70 puede descodificar uno o más elementos sintácticos representativos del valor del nivel de cada coeficiente significativo, por ejemplo, indicador_nivel_absoluto_coeficiente_mayor1, indicador_nivel_absoluto_coeficiente_mayor2 y resto_nivel_absoluto_coeficiente. En general, el indicador indicador_nivel_absoluto_coeficiente_mayor1 indica si el valor absoluto de un coeficiente significativo es mayor que 1, indicador_nivel_absoluto_coeficiente_mayor2_flag indica si el valor absoluto de un coeficiente significativo es mayor que 2, y resto_nivel_absoluto_coeficiente indica el valor absoluto de un coeficiente significativo menos 2.

La unidad de descodificación por entropía 70 también puede determinar la posición del último coeficiente significativo del bloque (por ejemplo, la TU) que se está regenerando. Más específicamente, la unidad de descodificación por entropía 70 puede identificar la posición (por ejemplo, en base a los elementos sintácticos codificados representativos de las coordenadas x - e y -) del último coeficiente significativo dentro de la TU asociada con el flujo de bits de vídeo codificado. En base a la identificación de la posición del último coeficiente significativo, la

unidad de descodificación por entropía 70 puede establecer los valores de los coeficientes restantes en la TU en orden de exploración a cero. Es decir, no es necesario que el descodificador de vídeo 30 reciba ningún elemento sintáctico para los coeficientes más allá del último coeficiente significativo y, además, puede inferir valores de 0 para estos coeficientes. Además, la unidad de descodificación por entropía 70 puede implementar una o más técnicas de esta divulgación para descodificar los bins de un valor binarizado que representan las coordenadas x- e y- de la posición del último coeficiente significativo utilizando una función, generalmente denotada por f(i), donde i corresponde a la posición del bin en el valor binarizado. En algunos ejemplos, la unidad de descodificación por entropía 70 puede descodificar datos codificados utilizando un contexto determinado para reproducir un valor para el bin, por ejemplo, "0" o "1" Aunque se describe como que corresponde a la posición del último coeficiente significativo, las técnicas de esta divulgación también se pueden aplicar a la descodificación por entropía de otros elementos sintácticos. Por ejemplo, la unidad de descodificación por entropía 70 puede descodificar por entropía varios elementos sintácticos, tales como elementos sintácticos para los coeficientes cuantificados enviados a la unidad de compensación de movimiento 72 y/o la unidad de intra-predicción 74, los elementos sintácticos representativos de los coeficientes de transformación cuantificados, y/o los valores que representan un último coeficiente significativo de la TU asociada con el flujo de bits de vídeo codificado, usando contextos determinados usando una o más funciones de los índices de bin de un valor representativo del elemento sintáctico correspondiente.

Por ejemplo, "Ctx_i" puede denotar el índice de contexto utilizado por la unidad de descodificación por entropía 70 para descodificar el i-ésimo bin en un valor binarizado que representa la posición del último coeficiente significativo, como se ha descrito anteriormente con respecto a las tablas 1-2 y 8-9. En este ejemplo, la unidad de descodificación por entropía 70 puede obtener el valor de Ctx_i usando la ecuación $Ctx_i = f(i)$, donde f(i) puede ser una función predefinida accesible a la unidad de descodificación por entropía 70 (por ejemplo, comunicada mediante el dispositivo de origen 12), o una función seleccionada por un usuario. Además, la unidad de descodificación por entropía 70 puede descodificar datos representativos de f(i), con el fin de utilizar los datos representativos de f(i) para obtener el valor de Ctx_i.

En algunos ejemplos, la unidad de descodificación por entropía 70 está configurada para determinar contextos para la descodificación de bins de datos que representan la posición del último coeficiente significativo utilizando las fórmulas (5)-(8) descritas anteriormente. Es decir, la unidad de descodificación por entropía 70 puede calcular f(i) de la forma siguiente: $Ctx_idx = desplazamiento + (i >> k)$. Por otra parte, la unidad de descodificación por entropía 70 puede obtener los valores del valor de desplazamiento y k utilizado en f(i) usando las siguientes ecuaciones:

$$desplazamiento = 3 * n + ((n + 1) >> 2),$$

$$k = (n + 3) >> 2,$$

y

$$n = (\log_2(\text{block_size}) - 2).$$

En otras implementaciones, la unidad de descodificación por entropía 70 puede establecer f(i) a una o más ecuaciones de ejemplo (1)-(4) y (9)-(12) en la descodificación del último coeficiente significativo de una TU representada por el flujo de bits de vídeo codificado. De esta manera, el descodificador de vídeo 30 y los componentes del mismo, tales como unidad de descodificación por entropía 70, pueden implementar las técnicas de esta divulgación para descodificar el último coeficiente significativo usando una o más funciones. Dichas funciones se pueden almacenar de manera más eficiente en la memoria del codificador de vídeo 20 y el descodificador de vídeo 30 que en tablas. Por lo tanto, las técnicas de esta divulgación pueden proporcionar codificadores de vídeo y descodificadores de vídeo que utilizan la memoria más eficientemente, por ejemplo, mediante la asignación de memoria que de otro modo se dedicaría a una tabla para otros datos, o mediante la reducción de la cantidad necesaria de memoria para un codificador de vídeo o un descodificador de vídeo:

Cuando el fragmento de vídeo se codifica como un fragmento intra-codificado (1), la unidad de intra-predicción 74 puede generar datos de predicción para un bloque de vídeo del fragmento de vídeo actual, en base a un modo de intra-predicción señalizada, 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, en base a los vectores de movimiento y a otros elementos sintácticos recibidos desde la unidad de descodificación por entropía 70. Los bloques predictivos pueden ser generados 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 por omisión, en base a las imágenes de referencia almacenadas en la memoria de tramas 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 para el bloque de vídeo actual que está siendo descodificado. 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), usada 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.

La unidad de compensación de movimiento 72 también puede realizar la interpolación en base a 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 fracciones de píxeles de bloques de referencia. En este caso, la unidad de compensación de movimiento 72 puede determinar los filtros de interpolación usados por el codificador de vídeo 20 a partir de los elementos sintácticos recibidos y usar los filtros de interpolación para generar bloques predictivos.

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 70. El proceso de cuantificación inversa puede incluir el uso de un parámetro de cuantificación QP_Y , calculado por el descodificador de vídeo 30 para 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.

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 generar bloques residuales en el dominio de píxeles.

Después de que la unidad de compensación de movimiento 72 genera el bloque predictivo para el bloque de vídeo actual, en base a los vectores de movimiento y a otros elementos sintácticos, el descodificador de vídeo 30 forma un bloque de vídeo descodificado sumando los bloques residuales procedentes de la unidad de transformación inversa 78 a 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. Otros filtros de bucle (ya sea en el bucle de codificación o después del bucle de codificación) también pueden ser usados para allanar 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 tramas 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.

De esta manera, el descodificador de vídeo 30 de la FIG. 3 representa un ejemplo de un descodificador de vídeo configurado para determinar un contexto para la codificación por entropía un bin de un valor indicativo de un último coeficiente significativo en un bloque de datos de vídeo usando una función de un índice del bin, y codificar el bin usando el contexto determinado. Además, el descodificador de vídeo 30 también representa un ejemplo de un descodificador de vídeo en la que la función produce un índice de contexto para el contexto, desplazando a la derecha el índice del bin un valor k y añadiendo el valor desplazado a la derecha a un valor de desplazamiento, en el que el valor de desplazamiento se determina según la fórmula $\text{desplazamiento} = 3 * n + ((n + 1) >> 2)$, en el que el valor k se determina según la fórmula $k = (n + 3) >> 2$, y en el que el valor n se determina según la fórmula $n = (\log_2(\text{block_size}) - 2)$.

La FIG. 4 es un diagrama de flujo que ilustra un procedimiento de ejemplo para la codificación de un bloque actual. El bloque actual puede comprender una CU actual o una parte de la CU actual. Aunque se describe con respecto al codificador de vídeo 20 (FIG. 1 y 2), se debe entender que otros dispositivos pueden configurarse para llevar a cabo un procedimiento similar al de la FIG. 4. Además, aunque el procedimiento de ejemplo de la FIG. 4 describe específicamente la codificación de los elementos sintácticos en relación con la posición del último coeficiente significativo de un bloque de vídeo utilizando estas técnicas, se debe entender que estas técnicas también pueden aplicarse a la codificación de otros elementos sintácticos.

En este ejemplo, el codificador de vídeo 20 inicialmente predice el bloque actual (150). Por ejemplo, el codificador de vídeo 20 puede calcular una o más unidades de predicción (PU) para el bloque actual. A continuación, el codificador de vídeo 20 puede calcular un bloque residual para el bloque actual, por ejemplo, para producir una unidad de transformación (TU) (152). Para calcular el bloque residual, el codificador de vídeo 20 puede calcular una diferencia entre el bloque no codificado original y el bloque predicho para el bloque actual. A continuación, el codificador de vídeo 20 puede transformar y cuantificar los coeficientes del bloque residual (154). A continuación, el codificador de vídeo 20 puede explorar los coeficientes de transformación cuantificados del bloque residual (156). Durante la exploración, o después de la exploración, el codificador de vídeo 20 puede codificar por entropía los coeficientes (158). Por ejemplo, el codificador de vídeo 20 puede codificar los coeficientes utilizando CAVLC o CABAC.

El codificador de vídeo 20 también puede determinar un valor para una posición de un último coeficiente significativo en la TU (160). El valor puede comprender, por ejemplo, un valor binarizado representativo de la posición del último coeficiente significativo, por ejemplo, como el descrito con respecto a la tabla 1 anterior. Un número máximo de bins del valor puede codificarse usando CABAC, mientras que otros bins que excedan el número máximo pueden codificarse por derivación, de nuevo como se ha descrito con respecto a la tabla 1. En particular, según las técnicas de esta divulgación, el codificador de vídeo 20 puede determinar contextos para bins del valor utilizando una función (162). Como se explicó anteriormente, los contextos pueden describir probabilidades de que los bins tengan un valor particular, por ejemplo, "0" o "1". La función puede corresponder a una de las funciones (1)-(12) descritas anteriormente, o a una función conceptualmente similar.

Con respecto a los ejemplos de funciones (5)-(8), el codificador de vídeo 20 puede determinar un contexto, ctx_idx , para un bin en la posición i de un valor binarizado representativo de una posición de un último coeficiente significativo, utilizando la fórmula desplazamiento + ($i >> k$), donde desplazamiento = $3 * n + ((n + 1) >> 2)$, $k = (n + 3) >> 2$ y $n = (\log_2(\text{block_size}) - 2)$. Es decir, el codificador de vídeo 20 puede iterar a través de cada bin que debe codificarse por entropía y ejecutar las funciones mostradas anteriormente para determinar un contexto para la codificación de un bin de la iteración actual. A continuación, el codificador de vídeo 20 puede codificar los bins del valor (por ejemplo, los bins que no superen el número máximo de bins), utilizando los contextos determinados (164). Asimismo, el codificador de vídeo 20 puede codificar por derivación cualquier bin restante del valor (166).

De esta manera, el procedimiento de la FIG. 4 representa un ejemplo de un procedimiento que incluye la determinación de un contexto para la codificación por entropía un bin de un valor indicativo de un último coeficiente significativo de un bloque de datos de vídeo usando una función de un índice del bin, y la codificación del bin utilizando el contexto determinado. Por otra parte, la función puede producir un índice de contexto para el contexto, desplazando a la derecha el índice del bin un valor k y añadiendo el valor desplazado a la derecha a un valor de desplazamiento, en el que el valor de desplazamiento se determina según la fórmula desplazamiento = $3 * n + ((n + 1) >> 2)$, en el que el valor k se determina según la fórmula $k = (n + 3) >> 2$, y en el que el valor de n se determina según la fórmula $n = (\log_2(\text{block_size}) - 2)$.

La FIG. 5 es un diagrama de flujo que ilustra un procedimiento de ejemplo para descodificar un bloque actual de datos de vídeo. El bloque actual puede comprender una CU actual o una parte de la CU actual. Aunque se describe con respecto al descodificador de vídeo 30 (FIG. 1 y 3), se debe entender que otros dispositivos pueden configurarse para llevar a cabo un procedimiento similar al de la FIG. 5. Además, aunque el procedimiento de ejemplo de la FIG. 4 describe específicamente la codificación de los elementos sintácticos en relación con la posición del último coeficiente significativo de un bloque de vídeo utilizando estas técnicas, se debe entender que estas técnicas también pueden aplicarse a la codificación de otros elementos sintácticos.

El descodificador de vídeo 30 puede predecir el bloque actual (200), por ejemplo, usando un modo de intra-predicción o inter-predicción para calcular un bloque predicho para el bloque actual. El descodificador de vídeo 30 también puede recibir datos codificados por entropía para el bloque actual, tales como los datos codificados por entropía para los coeficientes de un bloque residual correspondiente al bloque actual (202). El descodificador de vídeo 30 puede descodificar por entropía los datos codificados por entropía para reproducir los coeficientes del bloque residual (204).

Según las técnicas de esta divulgación, el descodificador de vídeo 30 puede recibir un valor codificado indicativo de una posición de un último coeficiente significativo en la TU (206). Un número máximo de bins del valor puede descodificarse usando CABAC, mientras que otros bins que excedan el número máximo de derivación pueden descodificarse por derivación, como se ha descrito con respecto a la tabla 1. En particular, según las técnicas de esta divulgación, el descodificador de vídeo 30 puede determinar contextos para bins del valor utilizando una función (208). Como se explicó anteriormente, los contextos pueden describir probabilidades de que los bins tengan un valor particular, por ejemplo, "0" o "1". La función puede corresponder a una de las funciones (1)-(12) descritas anteriormente, o a una función conceptualmente similar.

Con respecto a los ejemplos de funciones (5)-(8), el descodificador de vídeo 30 puede determinar un contexto, ctx_idx , para un bin en la posición i de un valor binarizado que se está descodificando, donde el valor binarizado es representativo de una posición de un último coeficiente significativo, utilizando la fórmula desplazamiento + ($i >> k$), donde desplazamiento = $3 * n + ((n + 1) >> 2)$, $k = (n + 3) >> 2$, y $n = (\log_2(\text{block_size}) - 2)$. Es decir, el descodificador de vídeo 30 puede descodificar iterativamente cada bin a descodificar por entropía y ejecutar las funciones mostradas anteriormente para determinar un contexto para la codificación de un bin de la iteración actual. A continuación, el descodificador de vídeo 30 puede descodificar los bins del valor (por ejemplo, los bins que no superen el número máximo de bins), utilizando los contextos determinados (210). Por ejemplo, el descodificador de vídeo 30 puede descodificar los datos codificados recibidos del codificador de vídeo 20 utilizando los contextos determinados para reproducir o de otro modo obtener los bins del valor. Del mismo modo, el descodificador de vídeo 30 puede descodificar por derivación cualquier bin restante del valor (212).

A continuación, el descodificador de vídeo 30 puede explorar inversamente los coeficientes reproducidos en base a

la posición del último coeficiente significativo (214), para crear un bloque de coeficientes de transformación cuantificados. Es decir, el descodificador de vídeo 30 puede colocar los coeficientes descodificados en la TU, comenzando en la posición del último coeficiente significativo, y proceder en un orden de exploración que en general corresponde al orden de exploración utilizado por el codificador. A continuación, el descodificador de vídeo 30 puede cuantificar inversamente y transformar inversamente los coeficientes para producir un bloque residual (216). El descodificador de vídeo 30 puede en última instancia descodificar el bloque actual combinando el bloque predicho y el bloque residual (218).

De esta manera, el procedimiento de la FIG. 5 representa un ejemplo de un procedimiento que incluye la determinación de un contexto para la codificación por entropía un bin de un valor indicativo de un último coeficiente significativo de un bloque de datos de vídeo usando una función de un índice del bin, y la codificación del bin utilizando el contexto determinado. Por otra parte, la función puede producir un índice de contexto para el contexto, desplazando a la derecha el índice del bin un valor k y añadiendo el valor desplazado a la derecha a un valor de desplazamiento, en el que el valor de desplazamiento se determina según la fórmula desplazamiento = $3 * n + ((n + 1) >> 2)$, en donde el valor de k se determina según la fórmula $k = (n + 3) >> 2$, y en donde el valor de n se determina según la fórmula $n = (\log_2(\text{block_size}) - 2)$.

Ha de 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 hebras, el procesamiento de interrupciones o múltiples procesadores, en lugar de secuencialmente.

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

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 puede denominarse debidamente un medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, un servidor u otra fuente remota 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, debe comprenderse 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, pero en lugar de eso están dirigidos a medios de almacenamiento tangibles no transitorios. Los discos, como se usan en el presente documento, incluyen el disco compacto (CD), el disco de láser, el disco óptico, el disco versátil digital (DVD), el disco flexible y el disco Blu-ray, donde algunos discos normalmente reproducen datos de manera magnética, mientras que otros discos reproducen los datos de manera óptica con láser. Las combinaciones de lo que antecede también deberían incluirse dentro del alcance de los medios legibles por ordenador.

Las instrucciones pueden ser ejecutadas por uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), matrices de puertas 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 cualquier estructura anterior 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 los módulos de hardware y/o software especializados configurados para la codificación y la descodificación, o incorporarse en un codec combinado. Igualmente, las técnicas pueden implementarse completamente en uno o más circuitos o elementos lógicos.

Las técnicas de esta divulgación se pueden implementar en una gran variedad de dispositivos o aparatos, incluyendo un auricular inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). Varios 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 la realización mediante diferentes unidades de hardware. Más bien, como se ha descrito anteriormente, pueden combinarse diversas unidades en una unidad de hardware de codec o ser proporcionadas por una colección de unidades de hardware inter-operativas, incluyendo uno o más procesadores, como se ha descrito anteriormente, junto con el software y/o firmware adecuado.

- 5
- 10

REIVINDICACIONES

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

5 la determinación (162; 208) de un contexto para la codificación por entropía de un bin de un valor indicativo de una posición del último coeficiente significativo de un bloque de datos de vídeo usando una función de un índice del bin, y la codificación (164; 210) del bin utilizando el contexto determinado; caracterizado por:

10 el uso de la función para producir un índice de contexto para el contexto desplazando a la derecha el índice del bin un valor k y añadiendo el valor desplazado a la derecha a un valor de desplazamiento, en el que el valor de desplazamiento se determina según la fórmula:

$$\text{desplazamiento} = 3*n + ((n + 1) \gg 2),$$

15 en donde el valor de k se determina según la fórmula:

$$k = (n+3) \gg 2$$

20 en donde el valor de n se determina según la fórmula:

$$n = (\log_2(\text{block_size})-2), \text{ y}$$

25 en donde el valor block_size comprende un valor indicativo del tamaño de bloque de la unidad de transformación.

2. El procedimiento, de acuerdo con la reivindicación 1, en el que la determinación del contexto comprende la ejecución de la función.

30 3. El procedimiento, de acuerdo con la reivindicación 1, en el que la función comprende una función lineal:

4. El procedimiento, de acuerdo con la reivindicación 1, que comprende además la recepción de la función de un usuario.

35 5. El procedimiento, de acuerdo con la reivindicación 1, que comprende además la recepción de datos sintácticos que definen la función.

6. El procedimiento, de acuerdo con la reivindicación 1, en el que la codificación del bin comprende descodificación por entropía (210) de los datos codificados utilizando el contexto determinado para reproducir un valor para el bin.

7. El procedimiento, de acuerdo con la reivindicación 1, en el que la codificación del bin comprende la codificación por entropía (158) del bin utilizando el contexto determinado.

45 8. Un dispositivo (20; 30) para codificar datos de vídeo, comprendiendo el dispositivo:

Medios (56; 70) para determinar un contexto para la codificación por entropía de un bin de un valor indicativo de una posición del último coeficiente significativo de un bloque de datos de vídeo usando una función de un índice del bin; y medios para codificar el bin utilizando el contexto determinado; caracterizado porque:

50 los medios para determinar un contexto están configurados para utilizar la función para producir un índice de contexto para el contexto desplazando a la derecha el índice del bin un valor k y añadiendo el valor desplazado a la derecha a un valor de desplazamiento, en donde el valor del desplazamiento se determina según la fórmula:

$$\text{desplazamiento} = 3*n + ((n + 1) \gg 2),$$

60 en donde el valor k se determina según la fórmula:

$$k = (n+3) \gg 2,$$

en donde el valor n se determina según la fórmula:

65 $n = (\log_2(\text{block_size})-2),$

y

en donde el valor `block_size` comprende un valor indicativo del tamaño de bloque de la unidad de transformación.

- 5 9. El dispositivo, de acuerdo con la reivindicación 8, en el que los medios para determinar el contexto comprenden medios para ejecutar la función.
10. El dispositivo, de acuerdo con la reivindicación 8, que comprende además medios para recibir datos sintácticos que definen la función.
- 10 11. El dispositivo, de acuerdo con la reivindicación 8, en el que los medios para codificar el bin están configurados para codificar el bin al menos en parte mediante la descodificación por entropía de los datos codificados utilizando el contexto determinado para reproducir un valor para el bin.
- 15 12. El dispositivo, de acuerdo con la reivindicación 8, en el que los medios para codificar el bin están configurados para codificar el bin al menos en parte mediante la codificación por entropía del bin utilizando el contexto determinado.
- 20 13. Un medio de almacenamiento legible por ordenador codificado con instrucciones que, cuando son ejecutadas, hacen que un procesador programable de un dispositivo informático lleve a cabo el procedimiento de acuerdo con cualquiera de las reivindicaciones 1 a 7.

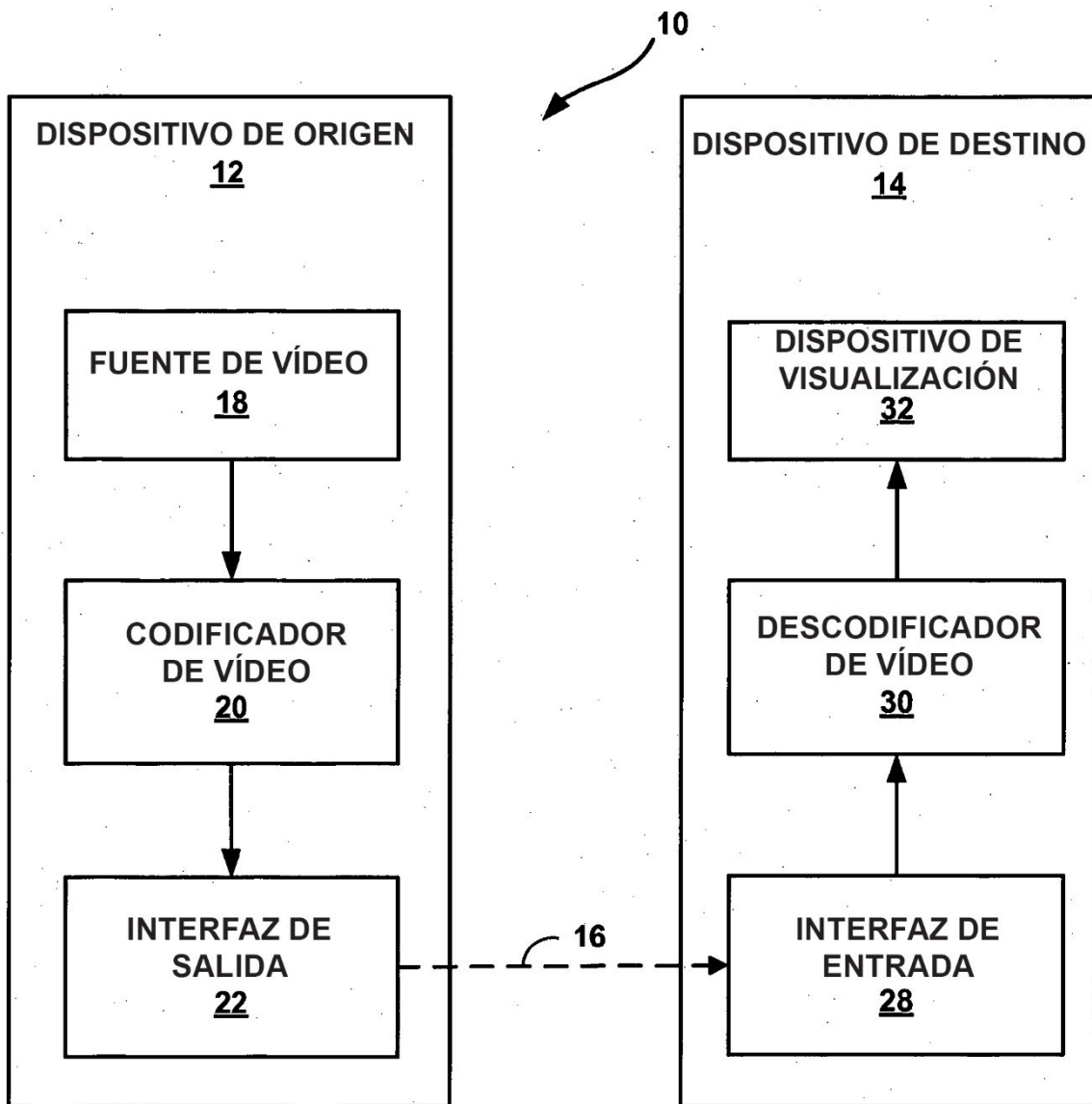


FIG. 1

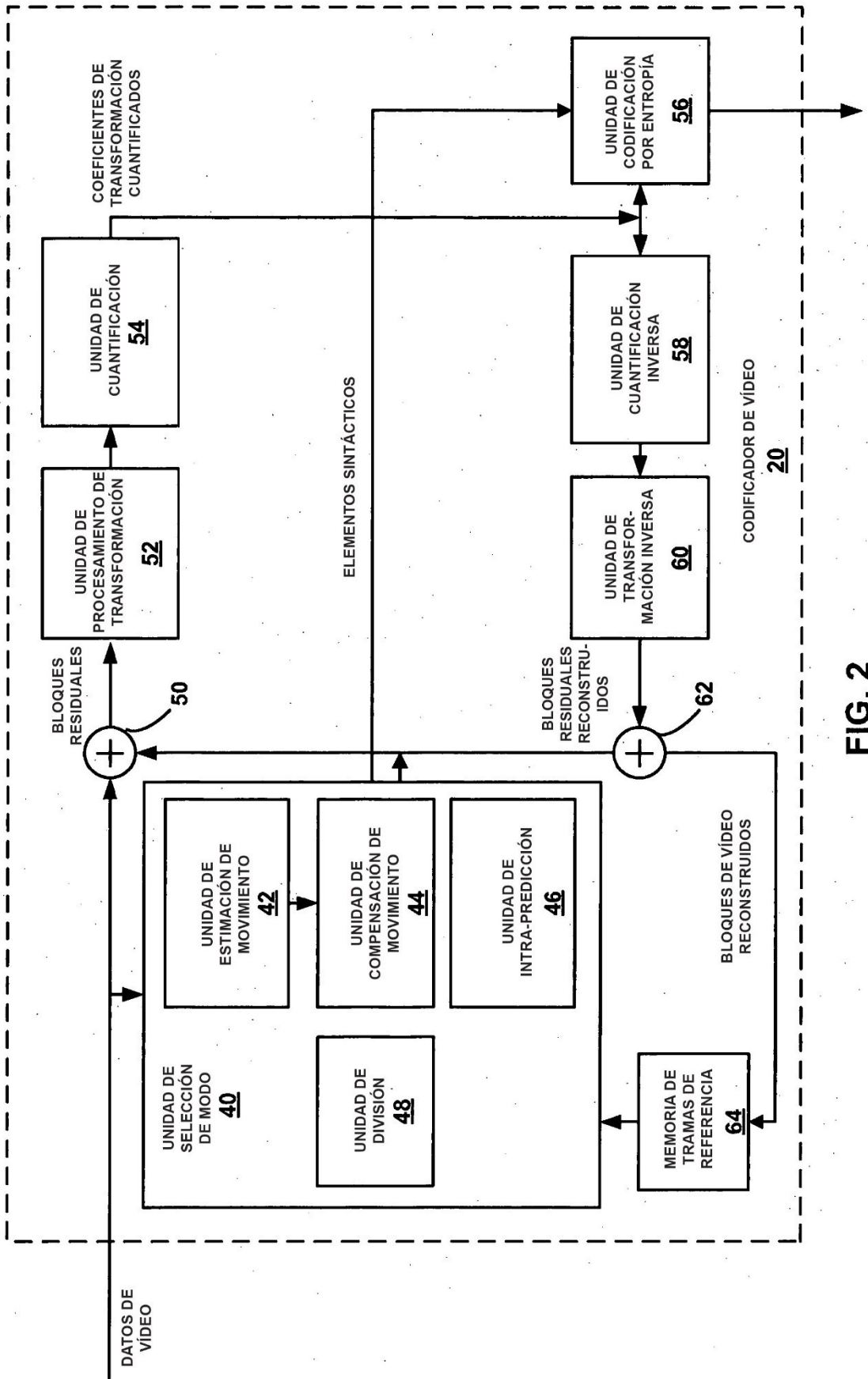


FIG. 2

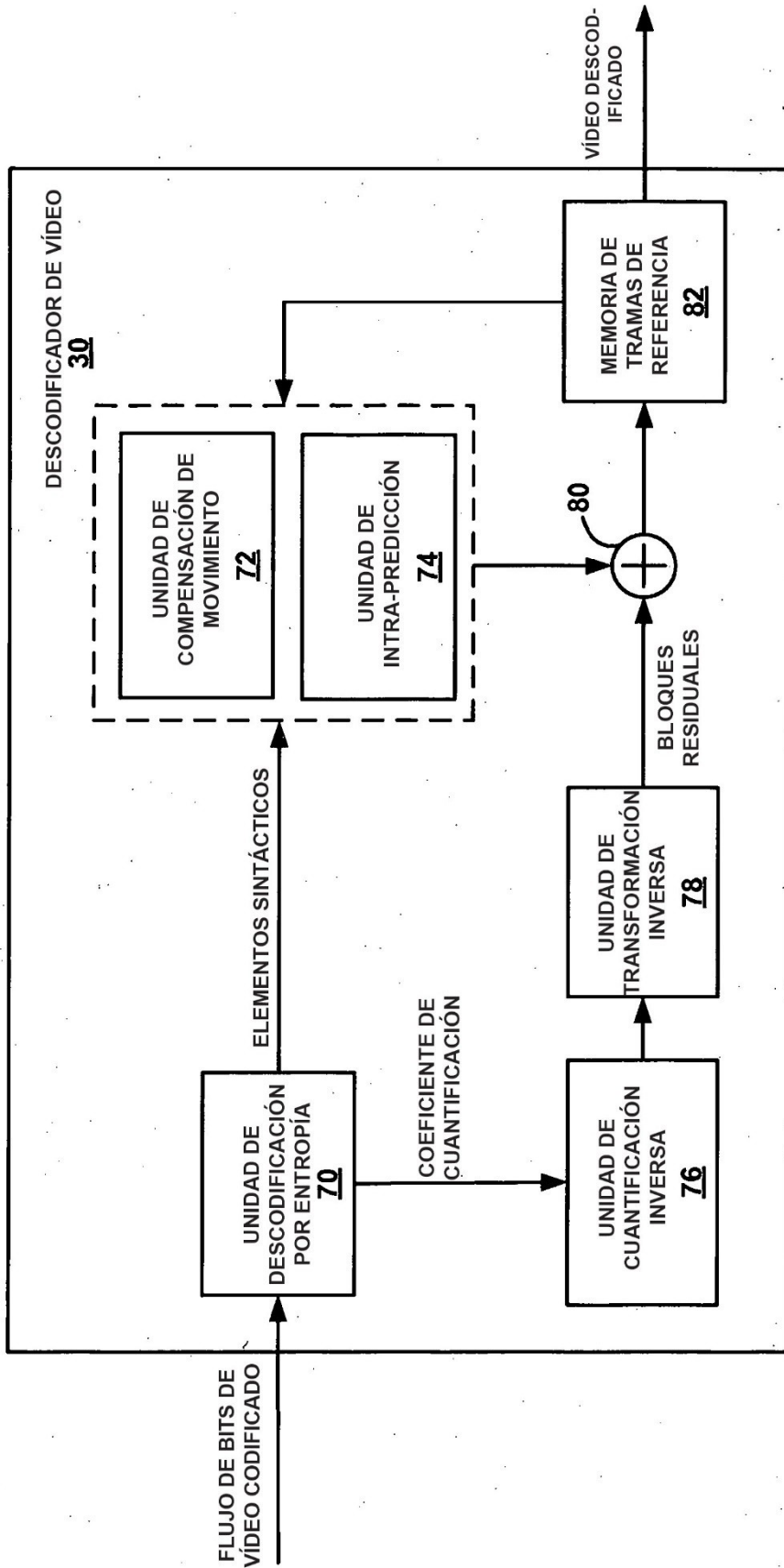


FIG. 3

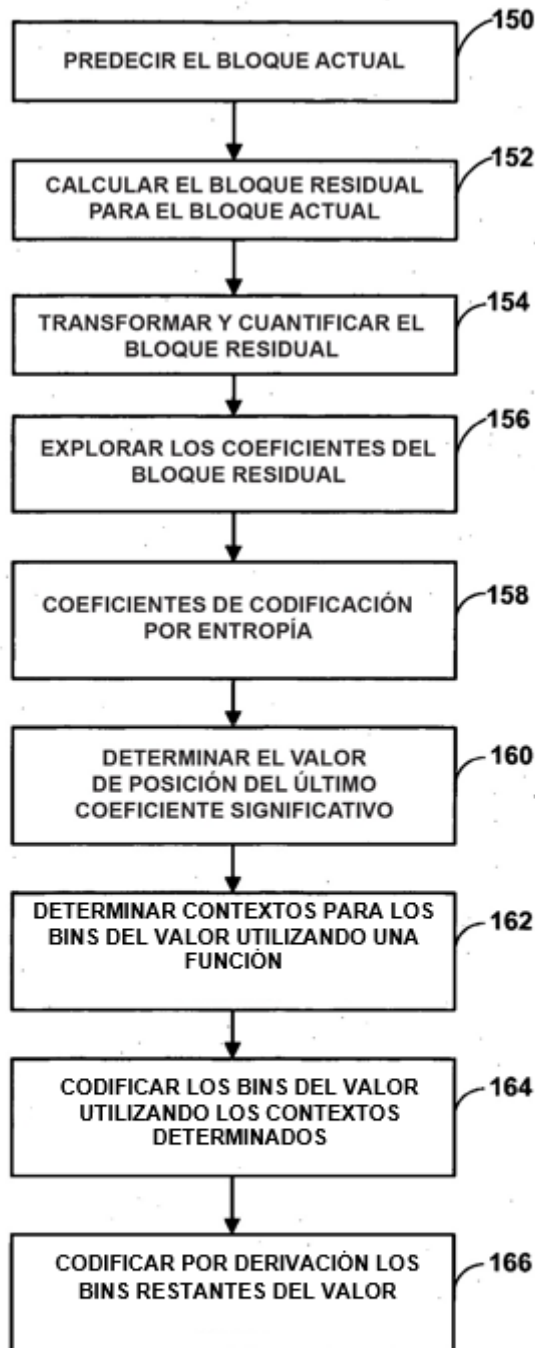


FIG. 4

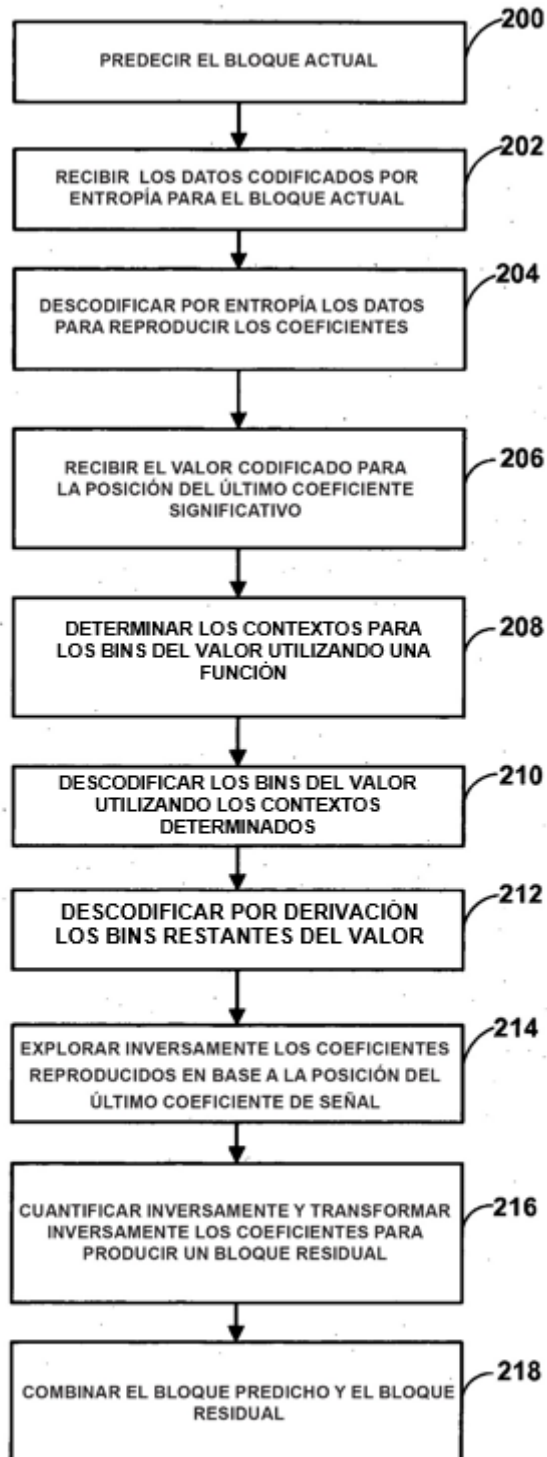


FIG. 5