

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 700 523**

51 Int. Cl.:

H04N 19/13 (2014.01)

H04N 19/18 (2014.01)

H04N 19/176 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **06.11.2012 PCT/US2012/063717**

87 Fecha y número de publicación internacional: **16.05.2013 WO13070610**

96 Fecha de presentación y número de la solicitud europea: **06.11.2012 E 12795670 (4)**

97 Fecha y número de publicación de la concesión europea: **10.10.2018 EP 2777163**

54 Título: **Optimización de contexto para la codificación de la posición del último coeficiente significativo**

30 Prioridad:

08.11.2011 US 201161557317 P

20.11.2011 US 201161561909 P

19.01.2012 US 201261588579 P

07.02.2012 US 201261596049 P

05.11.2012 US 201213669096

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
18.02.2019

73 Titular/es:

**VELOS MEDIA INTERNATIONAL LIMITED
(100.0%)**

**Unit 32, the Hyde Building The Park,
Carrickmines
Dublin 18, IE**

72 Inventor/es:

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

74 Agente/Representante:

MILTENYI , Peter

ES 2 700 523 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

Optimización de contexto para la codificación de la posición del último coeficiente significativo

Esta divulgación se refiere a la codificación de vídeo, y, más particularmente, a técnicas de codificación de coeficientes de transformada.

5 Antecedentes

Pueden incorporarse capacidades de vídeo digital en una amplia gama de dispositivos, incluyendo televisores digitales, sistemas de radiodifusión directa digital, sistemas de radiodifusión inalámbrica, agendas electrónicas (PDA), ordenadores portátiles o de sobremesa, ordenadores de tipo tableta, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, radiotelefonos móviles o vía satélite, los denominados "smartphones" (teléfonos inteligentes), dispositivos de vídeo/teleconferencia, dispositivos de emisión de vídeo en continuo, y similares. Los dispositivos de vídeo digital implementan técnicas de compresión de vídeo, tales como las descritas en las normas definidas por MPEG-2, MPEG-4, H.263 de ITU-T, H.264 de ITU-T/MPEG-4, Parte 10, Advanced Video Coding (Codificación de vídeo avanzada, AVC), la norma High Efficiency Video Coding (Codificación de vídeo de alta eficacia, HEVC) que está en desarrollo actualmente, y extensiones de tales normas. Los dispositivos pueden transmitir, recibir, codificar, decodificar y/o almacenar información de vídeo digital de manera más eficaz implementando tales técnicas de compresión de vídeo.

Las técnicas de compresión de vídeo realizan predicción espacial (intra-imagen) y/o predicción temporal (inter-imagen) para reducir o eliminar la redundancia inherente en las secuencias de vídeo. Para la codificación de vídeo basada en bloques, puede dividirse un segmento de vídeo (es decir, un fotograma de vídeo o una parte de un fotograma de vídeo) en bloques de vídeo, que también pueden denominarse bloques de árbol, unidades de codificación (CU) y/o nodos de codificación. Los bloques de vídeo en un segmento sometido a codificación intra (I) de una imagen se codifican usando predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen. Los bloques de vídeo en un segmento sometido a codificación inter (P o B) de una imagen pueden usar predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen o predicción temporal con respecto a muestras de referencia en otras imágenes de referencia. Las imágenes pueden denominarse fotogramas, y las imágenes de referencia pueden denominarse fotogramas de referencia.

La predicción espacial o temporal da como resultado un bloque predictivo para que se codifique un bloque. Los datos residuales representan diferencias de píxel entre el bloque original que va a codificarse y el bloque predictivo. Un bloque sometido a codificación inter 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 sometido a codificación intra se codifica según un modo de codificación intra y los datos residuales. Para una compresión adicional, los datos residuales pueden transformarse del dominio de píxeles a un dominio de transformada, dando como resultado coeficientes de transformada residuales, que entonces pueden cuantificarse. Los coeficientes de transformada cuantificados, dispuestos inicialmente en una matriz bidimensional, pueden someterse a exploración para producir un vector unidimensional de coeficientes de transformada, y puede aplicarse codificación de entropía para lograr incluso una mayor compresión.

Los siguientes documentos son bibliografía no de patentes relevante:

40 CHEUNG AUYEUNG ET AL: "Context reduction of the last transform position in JCTVC-D262 for CE11.1", n.º JCTVC-E344, 5º JCT-VC MEETING; 96º MPEG MEETING; 11-3-2011.

SOLE J ET AL: "Parallel Context Processing for the significance map in high coding efficiency", n.º JCTVC-D262, 4º JCT- VC MEETING; 95º MPEG MEETING; 16-01-2011.

Sumario

45 La invención está definida por el contenido de las reivindicaciones independientes.

En general, esta divulgación describe técnicas para codificar datos de vídeo. En particular, esta divulgación describe técnicas para codificar coeficientes de transformada.

En un ejemplo de la divulgación, un método de codificación de coeficientes de transformada comprende obtener una cadena binaria que indica la posición del último coeficiente significativo dentro de un bloque de coeficientes de transformada asociado con un bloque de vídeo; determinar un contexto para un índice binario de la cadena binaria basándose en el tamaño de bloque de vídeo, en el que el contexto se asigna a al menos dos índices binarios, en el que cada uno de los al menos dos índices binarios se asocian con tamaños de bloque de vídeo diferentes; y codificar la cadena binaria usando codificación aritmética binaria adaptativa según el contexto (Context-adaptive binary arithmetic coding, CABAC) basada al menos en parte en el contexto determinado.

5 En otro ejemplo de la divulgación, un método de decodificación de coeficientes de transformada comprende obtener una cadena binaria codificada que indica la posición del último coeficiente significativo dentro de un bloque de coeficientes de transformada asociado con un bloque de vídeo, en el que la cadena binaria codificada se codifica usando CABAC; determinar un contexto para un índice binario de la cadena binaria codificada basándose en el tamaño de bloque de vídeo, en el que el contexto se asigna a al menos dos índices binarios, en el que cada uno de los al menos dos índices binarios se asocian con tamaños de bloque de vídeo diferentes; y decodificar la cadena binaria codificada usando CABAC basada al menos en parte en el contexto determinado.

10 En otro ejemplo de la divulgación, un aparato configurado para codificar coeficientes de transformada en un procedimiento de codificación de vídeo comprende medios para obtener una cadena binaria que indica la posición del último coeficiente significativo dentro de un bloque de coeficientes de transformada asociado con unos medios de bloque de vídeo para determinar un contexto para un índice binario de la cadena binaria basándose en el tamaño de bloque de vídeo, en el que el contexto se asigna a al menos dos índices binarios, en el que cada uno de los al menos dos índices binarios se asocian con tamaños de bloque de vídeo diferentes; y medios para codificar la cadena binaria usando CABAC basada al menos en parte en el contexto determinado.

15 En otro ejemplo de la divulgación, un aparato configurado para decodificar coeficientes de transformada en un procedimiento de decodificación de vídeo comprende medios para obtener una cadena binaria codificada que indica la posición del último coeficiente significativo dentro de un bloque de coeficientes de transformada asociado con un bloque de vídeo, en el que la cadena binaria codificada se codifica usando CABAC; medios para determinar un contexto para un índice binario de la cadena binaria codificada basándose en el tamaño de bloque de vídeo, en el que el contexto se asigna a al menos dos índices binarios, en el que cada uno de los al menos dos índices binarios se asocian con tamaños de bloque de vídeo diferentes; y medios para decodificar la cadena binaria codificada usando CABAC basada al menos en parte en el contexto determinado.

20 En otro ejemplo de la divulgación, un dispositivo comprende un codificador de vídeo configurado para obtener una cadena binaria que indica la posición del último coeficiente significativo dentro de un bloque de coeficientes de transformada asociado con un bloque de vídeo; determinar un contexto para un índice binario de la cadena binaria basándose en el tamaño de bloque de vídeo, en el que el contexto se asigna a al menos dos índices binarios, en el que cada uno de los al menos dos índices binarios se asocian con tamaños de bloque de vídeo diferentes; y codificar la cadena binaria usando CABAC basada al menos en parte en el contexto determinado.

25 En otro ejemplo de la divulgación, un dispositivo comprende un decodificador de vídeo configurado para obtener una cadena binaria codificada que indica la posición del último coeficiente significativo dentro de un bloque de coeficientes de transformada asociado con un bloque de vídeo, en el que la cadena binaria codificada se codifica usando CABAC; determinar un contexto para un índice binario de la cadena binaria codificada basándose en el tamaño de bloque de vídeo, en el que el contexto se asigna a al menos dos índices binarios, en el que cada uno de los al menos dos índices binarios se asocian con tamaños de bloque de vídeo diferentes; y decodificar la cadena binaria codificada usando CABAC basada al menos en parte en el contexto determinado.

30 En otro ejemplo de la divulgación, un medio de almacenamiento legible por ordenador no transitorio tiene instrucciones almacenadas en el mismo que, tras su ejecución, hace que un dispositivo de codificación de vídeo obtenga una cadena binaria que indica la posición del último coeficiente significativo dentro de un bloque de coeficientes de transformada asociado con un bloque de vídeo; determine un contexto para un índice binario de la cadena binaria basándose en el tamaño de bloque de vídeo, en el que el contexto se asigna a al menos dos índices binarios, en el que cada uno de los al menos dos índices binarios se asocian con tamaños de bloque de vídeo diferentes; y codifique la cadena binaria usando CABAC basada al menos en parte en el contexto determinado.

35 En otro ejemplo de la divulgación, un medio de almacenamiento legible por ordenador no transitorio tiene instrucciones almacenadas en el mismo que, tras su ejecución, hace que un dispositivo de decodificación de vídeo obtenga una cadena binaria codificada que indica la posición del último coeficiente significativo dentro de un bloque de coeficientes de transformada asociado con un bloque de vídeo, en el que la cadena binaria codificada se codifica usando CABAC; determine un contexto para un índice binario de la cadena binaria codificada basándose en el tamaño de bloque de vídeo, en el que el contexto se asigna a al menos dos índices binarios, en el que cada uno de los al menos dos índices binarios se asocian con tamaños de bloque de vídeo diferentes; y decodifique la cadena binaria codificada usando CABAC basada al menos en parte en el contexto determinado.

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

Breve descripción de los dibujos

55 La figura 1 es un diagrama de bloques que ilustra un sistema de codificación y decodificación de vídeo a modo de ejemplo.

Las figuras 2A-2D ilustran órdenes de exploración de valor de coeficiente a modo de ejemplo.

La figura 3 ilustra un ejemplo de un mapa de significación en relación con un bloque de valores de coeficiente.

La figura 4 es un diagrama de bloques que ilustra un codificador de vídeo a modo de ejemplo que puede implementar las técnicas descritas en esta divulgación.

5 La figura 5 es un diagrama de bloques que ilustra un codificador de entropía a modo de ejemplo que puede implementar las técnicas descritas en esta divulgación.

La figura 6 es un diagrama de flujo que ilustra un ejemplo de codificación de un valor de cadena binaria que indica la posición del último coeficiente significativo según las técnicas de esta divulgación.

La figura 7 es un diagrama de bloques que ilustra un decodificador de vídeo a modo de ejemplo que puede implementar las técnicas descritas en esta divulgación.

10 La figura 8 es un diagrama de bloques que ilustra un decodificador de entropía a modo de ejemplo que puede implementar las técnicas descritas en esta divulgación.

La figura 9 es un diagrama de flujo que ilustra un ejemplo de decodificación de un valor de cadena binaria que indica la posición del último coeficiente significativo según las técnicas de esta divulgación.

Descripción detallada

15 En general, esta divulgación describe técnicas para codificar datos de vídeo. En particular, esta divulgación describe técnicas para codificar coeficientes de transformada en un procedimiento de codificación y/o decodificación de vídeo. En la codificación de vídeo basada en bloques, un bloque de coeficientes de transformada pueden disponerse en una matriz bidimensional (2D). Puede realizarse un procedimiento de exploración para rediseñar la matriz
 20 bidimensional (2D) de coeficientes de transformada en una matriz ordenada unidimensional (1D), es decir, un vector, de coeficientes de transformada. Pueden usarse uno o más elementos sintácticos para indicar la posición del último coeficiente significativo (es decir, coeficiente no nulo) dentro del bloque de coeficientes de transformada basándose en un orden de exploración. La posición del último coeficiente significativo puede usarse por un codificador de vídeo para optimizar la codificación de los coeficientes de transformada. Asimismo, un decodificador de vídeo puede usar la posición del último coeficiente significativo para optimizar la decodificación de coeficientes de transformada. Por
 25 tanto, es deseable codificar eficazmente el uno o más elementos sintácticos que indican la posición del último coeficiente significativo.

Esta divulgación describe técnicas para codificar el uno o más elementos sintácticos que indican la posición del último coeficiente significativo. En algunos ejemplos, la totalidad o una parte de los elementos sintácticos que indican la posición del último coeficiente significativo pueden someterse a codificación de entropía según cualquiera de las
 30 siguientes técnicas: codificación de longitud variable adaptativa según el contexto (Context Adaptive Variable Length Coding, CAVLC), codificación aritmética binaria adaptativa según el contexto (CABAC), codificación de entropía de división de intervalo de probabilidad (Probability Interval Entropy Coding, PIPE), o similares. En las técnicas de codificación de entropía que utilizan índices binarios que también pueden denominarse “bins” o “índices de bin”) y asignaciones de contexto, puede utilizarse una asignación de contexto habitual para bins para tamaños de
 35 bloque de transformada (TU) diferentes y/o componentes de color diferentes. De esta manera, puede reducirse el número total de contextos. Al reducir el número total de contextos, un codificador de vídeo y/o decodificador de vídeo pueden codificar de manera más eficaz elementos sintácticos que indican la posición del último coeficiente significativo, dado que es necesario almacenar menos contextos.

La figura 1 es un diagrama de bloques que ilustra un sistema de codificación y decodificación de vídeo 10 a modo de ejemplo que puede configurarse para utilizar técnicas para codificar coeficientes de transformada según ejemplos de esta divulgación. Tal como se muestra en la figura 1, el sistema 10 incluye un dispositivo fuente 12 que transmite vídeo codificado a un dispositivo de destino 14 mediante un canal de comunicación 16. También pueden almacenarse datos de vídeo codificados en un medio de almacenamiento 34 o un servidor de archivos 36 y el dispositivo de destino 14 puede acceder a ellos según se prefiera. Cuando se almacenan en un medio de
 45 almacenamiento o un servidor de archivos, el codificador de vídeo 20 puede proporcionar datos de vídeo codificados a otro dispositivo, tal como una interfaz de red, un disco compacto (CD), un dispositivo de instalación de estampado o de grabación de discos Blu-ray o discos de vídeo digital (DVD), u otros dispositivos, para almacenar los datos de vídeo codificados en el medio de almacenamiento. Asimismo, un dispositivo independiente del decodificador de vídeo 30, tal como una interfaz de red, lector de CD o DVD, o similar, puede recuperar datos de vídeo codificados de un medio de almacenamiento y proporcionar los datos recuperados al decodificador de vídeo 30.
 50

El dispositivo fuente 12 y el dispositivo de destino 14 pueden comprender cualquiera de una amplia variedad de dispositivos, incluyendo ordenadores de sobremesa, ordenadores tipo *notebook* (es decir, portátiles), ordenadores de tipo tableta, decodificadores, terminales de teléfonos tales como los denominados teléfonos inteligentes, televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, o similares. En muchos casos, tales dispositivos pueden estar equipados para comunicación inalámbrica. Por tanto, el canal de comunicación 16 puede comprender un canal inalámbrico, un canal por cable, o una combinación de canales inalámbricos y por cable adecuados para la transmisión de datos de vídeo codificados. De manera similar, el
 55

dispositivo de destino 14 puede acceder al servidor de archivos 36 mediante cualquier conexión de datos convencional, 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 archivos.

5 Pueden aplicarse técnicas para codificar coeficientes de transformada, según ejemplos de esta divulgación, a la codificación de vídeo como soporte de cualquiera de una variedad de aplicaciones multimedia, tales como radiodifusiones de televisión sin hilo, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo de emisión en continuo, por ejemplo, a través de Internet, codificación de vídeo digital para su almacenamiento en un medio de almacenamiento de datos, decodificación de vídeo digital almacenado en un
10 medio de almacenamiento de datos, u otras aplicaciones. En algunos ejemplos, el sistema 10 puede configurarse para soportar transmisión de vídeo uni o bidireccional para soportar aplicaciones tales como emisión de vídeo en continuo, reproducción de vídeo, radiodifusión de vídeo y/o videotelefonía.

En el ejemplo de la figura 1, el dispositivo fuente 12 incluye una fuente de vídeo 18, un codificador de vídeo 20, un modulador/demodulador 22 y un transmisor 24. En el dispositivo fuente 12, la fuente de vídeo 18 puede incluir una
15 fuente tal como un dispositivo de captura de vídeo, tal como una cámara de vídeo, un archivo de vídeo que contiene vídeo capturado previamente, una interfaz de señal de vídeo para recibir vídeo de un proveedor de contenido de vídeo, y/o un sistema de gráficos por ordenador para generar datos de gráficos por ordenador como el vídeo fuente, o una combinación de tales fuentes. Como ejemplo, si la fuente de vídeo 18 es una cámara de vídeo, el dispositivo fuente 12 y el dispositivo de destino 14 pueden formar los denominados teléfonos con cámara o teléfonos con vídeo.
20 Sin embargo, las técnicas descritas en esta divulgación pueden ser aplicables a la codificación de vídeo en general, y pueden aplicarse a aplicaciones inalámbricas y/o por cable, o una aplicación en la que se almacenen datos de vídeo codificados en un disco local.

El vídeo capturado, capturado previamente o generado por ordenador puede codificarse por el codificador de vídeo 20. La información de vídeo codificada puede modularse por el módem 22 según una norma de comunicación, tal
25 como un protocolo de comunicación inalámbrica, y transmitirse al dispositivo de destino 14 a través del transmisor 24. El módem 22 puede incluir diversos mezcladores, filtros, amplificadores u otros componentes diseñados para la modulación de señal. El transmisor 24 puede incluir circuitos diseñados para transmitir datos, que incluyen amplificadores, filtros, y una o más antenas.

El vídeo capturado, capturado previamente o generado por ordenador que se codifica por el codificador de vídeo 20 también puede almacenarse en un medio de almacenamiento 34 o un servidor de archivos 36 para su consumo
30 posterior. El medio de almacenamiento 34 puede incluir discos Blu-ray, DVD, CD-ROM, memoria *flash*, u cualquier otro medio de almacenamiento digital adecuado para almacenar vídeo codificado. Entonces el dispositivo de destino 14 puede acceder al vídeo codificado almacenado en el medio de almacenamiento 34 para su decodificación y reproducción.

El servidor de archivos 36 puede ser cualquier tipo de servidor que pueda almacenar vídeo codificado y transmitir ese vídeo codificado al dispositivo de destino 14. Servidores de archivo a modo de ejemplo incluyen un servidor web (por ejemplo, para un sitio web), un servidor FTP, dispositivos de almacenamiento conectados en red (NAS), una
35 unidad de disco local, o cualquier otro tipo de dispositivo que pueda almacenar datos de vídeo codificados y transmitirlos a un dispositivo de destino. La transmisión de datos de vídeo codificados procedentes del servidor de archivos 36 puede ser una transmisión de emisión en continuo, una transmisión de descarga, o una combinación de ambas. El dispositivo de destino 14 puede acceder al servidor de archivos 36 mediante cualquier conexión de datos convencional, incluyendo una conexión a Internet. Esta puede incluir un canal inalámbrico (por ejemplo, una conexión Wi-Fi), una conexión por cable (por ejemplo, DSL, módem de cable, Ethernet, USB, etc.), o una
40 combinación de ambas que sea adecuada para acceder a datos de vídeo codificados almacenados en un servidor de archivos.
45

El dispositivo de destino 14, en el ejemplo de la figura 1, incluye un receptor 26, un módem 28, un decodificador de vídeo 30 y un dispositivo de visualización 32. El receptor 26 del dispositivo de destino 14 recibe información por el canal 16, y el módem 28 demodula la información para producir un flujo de bits demodulado para el decodificador de vídeo 30. La información comunicada por el canal 16 puede incluir diversa información sintáctica generada por el
50 codificador de vídeo 20 para su uso por el decodificador de vídeo 30 en la decodificación de datos de vídeo. Tal sintaxis también puede incluirse con los datos de vídeo codificados almacenados en el medio de almacenamiento 34 o el servidor de archivos 36. Cada uno del codificador de vídeo 20 y el decodificador de vídeo 30 pueden formar parte de un codificador-decodificador (CODEC) respectivo que puede codificar o decodificar datos de vídeo.

El dispositivo de visualización 32 puede estar integrado con, o ser externo a, el dispositivo de destino 14. En algunos ejemplos, el dispositivo de destino 14 puede incluir un dispositivo de visualización integrado y estar configurado también para interconectarse con un dispositivo de visualización externo. En otros ejemplos, el dispositivo de destino 14 puede ser un dispositivo de visualización. En general, el dispositivo de visualización 32 presenta visualmente los
55 datos de vídeo codificados a un usuario, y puede comprender cualquiera de una variedad de dispositivos de visualización tales como una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodo orgánico emisor de luz (OLED), u otro tipo de dispositivo de visualización.
60

En el ejemplo de la figura 1, el canal de comunicación 16 puede comprender cualquier medio de comunicación inalámbrica o por cable, tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión física, o cualquier combinación de medios inalámbricos y por cable. El canal de comunicación 16 puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área amplia o una red global tal como Internet.

5 El canal de comunicación 16 representa generalmente cualquier medio de comunicación adecuado, o colección de diferentes medios de comunicación, para transmitir datos de vídeo desde el dispositivo fuente 12 hasta el dispositivo de destino 14, incluyendo cualquier combinación adecuada de medios por cable o inalámbricos. El canal de comunicación 16 puede incluir encaminadores, conmutadores, estaciones base, o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo fuente 12 hasta el dispositivo de destino 14.

10 El codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar según una norma de compresión de vídeo, tal como la norma de High Efficiency Video Coding (HEVC) que está desarrollando actualmente el Equipo de colaboración conjunta sobre codificación de vídeo (JCT-VC) del Grupo de expertos de codificación de vídeo (Video Coding Experts Group, VCEG) de ITU-T y el Grupo de expertos de imágenes en movimiento (MPEG) de ISO/IEC, y puede adaptarse al modelo de prueba (HM) de HEVC. El codificador de vídeo 20 y el decodificador de vídeo 30

15 pueden funcionar según un borrador reciente de la norma HEVC, denominado "Borrador de trabajo 5 de HEVC" o "WD5", se describe en el documento JCTVC-G1103, Bross *et al.*, "WD5: Working Draft 5 of High efficiency video coding (HEVC)", Equipo de colaboración conjunta sobre codificación de vídeo (JCT-VC) de ITU-T SG16 WP3 y JTC1/SC29/WG11 de ISO/IEC, 7º encuentro: Ginebra, Suiza, noviembre de 2012. Además, otro borrador de trabajo reciente de HEVC, el Borrador de trabajo 7, se describe en el documento HCTVC-11003, Bross *et al.*, "High Efficiency Video Coding (HEVC) Text Specification Draft 7", Equipo de colaboración conjunta sobre codificación de vídeo (JCT-VC) de ITU-T SG16 WP3 y JTC1/SC29/WG11 de ISO/IEC, 9º encuentro: Ginebra, Suiza, 27 de abril de 2012 - 7 de mayo de 2012. Alternativamente, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar según otras normas patentadas o industriales, tales como la norma H.264 de ITU-T, denominada

20 alternativamente MPEG-4, parte 10, Codificación de vídeo avanzada (AVC), o extensiones de tales normas. Sin embargo, las técnicas de esta divulgación, no se limitan a ninguna norma de codificación particular. Otros ejemplos incluyen MPEG-2 y H.263 de ITU-T.

Aunque no se muestra en la figura 1, en algunos aspectos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden estar integrados cada uno con un codificador y decodificador de audio, y puede incluir unidades de MUX-DEMUX apropiadas, u otro hardware y software, para gestionar la codificación tanto de audio como de vídeo en un

30 flujo de datos común o flujos de datos independientes. Si es aplicable, en algunos ejemplos, las unidades de MUX-DEMUX pueden ser conformes al protocolo de multiplexor H.223 de ITU, u otros protocolos tales como el protocolo de datagramas de usuario (UDP).

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementarse cada uno como una variedad de conjuntos de circuito de codificador adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados para aplicaciones específicas (ASIC), matrices de puertas programables *in situ* (FPGA), lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. Cuando las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio legible por ordenador no transitorio adecuado, y ejecutar las instrucciones en hardware usando uno o más procesadores para realizar las técnicas de esta divulgación. Cada uno del codificador de vídeo 20 y el decodificador

40 de vídeo 30 puede estar incluido en una o más codificadores o decodificadores, cualquiera de los cuales puede estar integrado como parte de un codificador/decodificador combinado (CODEC) en un dispositivo respectivo.

El codificador de vídeo 20 puede implementar cualquiera o todas las técnicas de esta divulgación para codificar coeficientes de transformada en un procedimiento de codificación de vídeo. Asimismo, el decodificador de vídeo 30 puede implementar cualquiera o todas estas técnicas para codificar coeficientes de transformada en un

45 procedimiento de codificación de vídeo. Un codificador de vídeo, tal como se describe en esta divulgación, puede referirse a un codificador de vídeo o a un decodificador de vídeo. De manera similar, una unidad de codificación de vídeo puede referirse a un codificador de vídeo o a un decodificador de vídeo. Asimismo, la codificación de vídeo puede referirse a codificación de vídeo o decodificación de vídeo.

El codificador de vídeo 20 y el decodificador de vídeo 30 de la figura 1 representan ejemplos de codificadores de vídeo configurados para obtener una cadena binaria que indica la posición del último coeficiente significativo dentro de un bloque de vídeo, determinar un contexto para un índice binario de la cadena binaria basándose en el tamaño de bloque de vídeo, en el que el contexto se asigna a al menos dos índices binarios, en el que cada uno de los al menos dos índices binarios se asocian con tamaños de bloque de vídeo diferentes; y codificar la cadena binaria usando codificación aritmética binaria adaptativa según el contexto (CABAC) basada al menos en parte en el

50 contexto determinado.

Para una codificación de vídeo según la norma HEVC que está en desarrollo actualmente, un fotograma de vídeo puede dividirse en unidades de codificación. Una unidad de codificación (CU) se refiere generalmente a una región de imagen que sirve como unidad básica a la que se aplican diversas herramientas de codificación para compresión de vídeo. Una CU tiene habitualmente una componente de luminancia, indicada como Y, y dos componentes de croma, indicadas como U y V. Dependiendo del formato de muestreo de vídeo, el tamaño de las componentes U y V, en términos de número de muestras, puede ser el mismo o diferente del tamaño de la componente Y. Una CU es

60

normalmente cuadrada, y puede considerarse que es similar a un denominado macrobloque, por ejemplo, con arreglo a otras normas de codificación de vídeo tales como H.264 de ITU-T. La codificación según algunos de los aspectos propuestos actualmente de la norma HEVC en desarrollo se describirá en esta solicitud con propósitos ilustrativos. Sin embargo, las técnicas descritas en esta divulgación pueden ser útiles para otros procedimientos de codificación de vídeo, tales como los definidos según H.264 o u otros procedimientos de codificación de vídeo normativos o patentados. Los esfuerzos de normalización de HEVC se basan en un modelo de dispositivo de codificación de vídeo denominado modelo de prueba (HM) de HEVC. El HM supone varias capacidades de dispositivos de codificación de vídeo con respecto a dispositivos según, por ejemplo, H.264 de ITU-T/AVC. Por ejemplo, mientras que H.264 proporciona nueve modos de codificación de predicción intra, HM proporciona hasta treinta y cuatro modos de codificación de predicción intra.

Una secuencia de vídeo incluye normalmente una serie de imágenes o fotogramas 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, una cabecera de una o más de las imágenes, o en otro lugar, que describen un número de imágenes incluidas en el GOP. Cada segmento de una imagen puede incluir datos sintácticos de segmento que describen un modo de codificación para el segmento respectivo. El codificador de vídeo 20 funciona normalmente en bloques de vídeo dentro de segmentos de vídeo individuales para codificar los datos de vídeo. Un bloque de vídeo puede incluir una o más TU o PU que corresponden a un nodo de codificación dentro de una CU. El bloque de vídeos puede tener tamaños fijos o variables, y pueden diferir en tamaño según una norma de codificación especificada.

Según el HM, una CU puede incluir una o más unidades de predicción (PU) y/o una o más unidades de transformada (TU). Los datos sintácticos dentro de un flujo de bits pueden definir una unidad de codificación de mayor tamaño (Largest Coding Unit, LCU), que es la CU de mayor tamaño en términos del número de píxeles. En general, una CU tiene un propósito similar a un macrobloque de H.264, excepto porque una CU no tiene una distinción de tamaño. Por tanto, una CU puede dividirse en CU secundarias. En general, las referencias en esta divulgación a una CU pueden referirse a una unidad de codificación de mayor tamaño de una imagen o una CU secundaria de una LCU. Una LCU puede dividirse en CU secundarias, y cada CU secundaria puede dividirse adicionalmente en CU secundarias. Los datos sintácticos para un flujo de bits pueden definir el número máximo de veces que puede dividirse una LCU, denominado profundidad de CU. Por consiguiente, un flujo de bits también puede definir una unidad de codificación de menor tamaño (Smallest Coding Unit, SCU). Esta divulgación también usa el término "bloque" o "parte" para referirse a cualquiera de una CU, PU o TU. En general, "parte" puede referirse a cualquier subconjunto de un fotograma de vídeo.

Una LCU puede estar asociada con una estructura de datos de árbol cuaternario. En general, una estructura de datos de árbol cuaternario incluye un nodo por CU, en el que un nodo raíz corresponde a la LCU. Si se divide una CU en cuatro CU secundarias, el nodo correspondiente a la CU incluye cuatro nodos hoja, cada uno de los cuales corresponde a una de las CU secundarias. Cada nodo de la estructura de datos de árbol cuaternario puede proporcionar datos sintácticos para la CU correspondiente. Por ejemplo, un nodo en el árbol cuaternario puede incluir una bandera de división, que indica si la CU correspondiente al nodo se divide en CU secundarias. Los elementos sintácticos para una CU pueden definirse de manera recursiva, y pueden depender de si la CU se divide en CU secundarias. Si una CU no se divide adicionalmente, se denomina CU hoja. En esta divulgación, 4 CU secundarias de una CU hoja también se denominarán CU hoja aunque no haya ninguna división explícita de la CU hoja original. Por ejemplo, si una CU con un tamaño de 16x16 no se divide adicionalmente, las cuatro CU secundarias de 8x8 también se denominarán CU hoja aunque nunca se haya dividido la CU de 16x16.

Además, las TU de CU hoja también pueden asociarse con estructuras de datos de árbol cuaternario respectivas. Es decir, una CU hoja puede incluir un árbol cuaternario que indica cómo se divide la CU hoja en TU. Esta divulgación se refiere al árbol cuaternario que indica cómo se divide una LCU como árbol cuaternario de CU y el árbol cuaternario que indica cómo se divide una CU hoja en TU como árbol cuaternario de TU. El nodo raíz de un árbol cuaternario de TU corresponde generalmente a una CU hoja, mientras que el nodo raíz de un árbol cuaternario de CU corresponde generalmente a una LCU. Las TU del árbol cuaternario de TU que no se dividen se denominan TU hoja.

Una CU hoja puede incluir una o más unidades de predicción (PU). En general, una PU representa la totalidad o una parte de la CU correspondiente, y puede incluir datos para recuperar una muestra de referencia para la PU. Por ejemplo, cuando se codifica la PU en modo inter, la PU puede incluir datos que definen un vector de movimiento para la PU. Los datos que definen el vector de movimiento pueden describir, por ejemplo, una componente horizontal del vector de movimiento, una componente vertical del vector de movimiento, una resolución para el vector de movimiento (por ejemplo, precisión de un cuarto de píxel o precisión de un octavo de píxel), un fotograma de referencia al que apunta el vector de movimiento, y/o una lista de referencia (por ejemplo, lista 0 o lista 1) para el vector de movimiento. Los datos para la CU hoja que define la(s) PU también pueden describir, por ejemplo, la división de la CU en una o más PU. Los modos de división pueden diferir dependiendo de si la CU no está codificada, está codificada mediante el modo de predicción intra, o está codificada mediante el modo de predicción inter. Para la codificación intra, una PU puede tratarse de la misma manera que una unidad de transformada hoja descrita a continuación.

Como ejemplo, el HM soporta la predicción en diversos tamaños de PU. Suponiendo que el tamaño de una CU particular es de $2N \times 2N$, el HM soporta predicción intra en tamaños de PU de $2N \times 2N$ o $N \times N$, y predicción intra en tamaños de PU simétricos de $2N \times 2N$, $2N \times N$, $N \times 2N$ o $N \times N$. El HM también soporta división asimétrica para predicción inter 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 se divide, mientras que la otra dirección se divide en el 25% y el 75%. La parte de la CU correspondiente a la división del 25% se indica mediante una "n" seguida por una indicación de "arriba", "abajo", "izquierda" o "derecha". Por tanto, por ejemplo, " $2N \times nU$ " se refiere a una CU de $2N \times 2N$ que se divide horizontalmente con una PU de $2N \times 0,5N$ por encima y una PU de $2N \times 1,5N$ por debajo.

En esta divulgación, " $N \times N$ " y " N por N " pueden usarse de manera intercambiable para referirse a las dimensiones de píxeles de un bloque de vídeo en términos de dimensiones vertical y horizontal, por ejemplo, 16×16 píxeles o 16 por 16 píxeles. En general, un bloque de 16×16 tendrá 16 píxeles en una dirección vertical ($y = 16$) y 16 píxeles en una dirección horizontal ($x = 16$). Asimismo, un bloque de $N \times N$ generalmente tiene N píxeles en una dirección vertical y N píxeles en una dirección horizontal, donde N representa un valor de número entero no negativo. Los píxeles en un bloque pueden disponerse en filas y columnas. Además, no es necesario que los bloques tengan necesariamente el mismo número de píxeles en la dirección horizontal que en la dirección vertical. Por ejemplo, los bloques pueden comprender $N \times M$ píxeles, donde M no es necesariamente igual a N .

Para codificar un bloque (por ejemplo, una unidad de predicción de datos de vídeo), en primer lugar se deriva un elemento de predicción para el bloque. El elemento de predicción, también denominado bloque predictivo, puede derivarse o bien mediante predicción intra (I) (es decir, predicción espacial) o bien mediante predicción inter (P o B) (es decir, predicción temporal). Por tanto, algunas unidades de predicción pueden someterse a codificación intra (I) usando predicción espacial con respecto a muestras de referencia en bloques de referencia vecinos en el mismo fotograma (o segmento), y otras unidades de predicción pueden someterse a codificación inter de manera unidireccional (P) o someterse a codificación inter de manera bidireccional (B) con respecto a bloques de muestras de referencia en otros fotogramas (o segmentos) codificados previamente. En cada caso, las muestras de referencia pueden usarse para formar un bloque predictivo para que se codifique un bloque.

Tras la identificación de un bloque predictivo, se determina la diferencia entre el bloque de datos de vídeo original y su bloque predictivo. Esta diferencia puede denominarse datos residuales de predicción, e indica las diferencias de píxel entre los valores de píxel en el bloque que va a codificarse y los valores de píxel en el bloque predictivo seleccionado para representar el bloque codificado. Para lograr una mejor compresión, los datos residuales de predicción pueden transformarse, por ejemplo, usando una transformada discreta del coseno (DCT), una transformada entera, una transformada de Karhunen-Loeve (K-L) u otra transformada.

Los datos residuales en un bloque de transformada, tal como una TU, pueden disponerse en una matriz bidimensional (2D) de valores de diferencia de píxel que residen en el dominio espacial de píxeles. Una transformada convierte los valores de píxeles residuales en una matriz bidimensional de coeficientes de transformada en un dominio de transformada, tal como un dominio de frecuencia. Para una compresión adicional, los coeficientes de transformada pueden cuantificarse antes de la codificación de entropía. Entonces, un codificador de entropía aplica codificación de entropía, tal como CAVLC, CABAC, PIPE, o similar, a los coeficientes de transformada cuantificados.

Para someter a codificación de entropía un bloque de coeficientes de transformada cuantificados, se realiza habitualmente un procedimiento de exploración de modo que se redispone en un bloque la matriz bidimensional (2D) de coeficientes de transformada cuantificados, según un orden de exploración particular, para dar una matriz unidimensional (1D) ordenada, es decir, un vector, de coeficientes de transformada. Entonces se aplica codificación de entropía al vector de coeficientes de transformada. La exploración de los coeficientes de transformada cuantificados en una unidad de transformada serializa la matriz 2D de coeficientes de transformada para el codificador de entropía. Puede generarse un mapa de significación para indicar las posiciones de coeficientes significativos (es decir, no nulos). La exploración puede aplicarse a niveles de exploración de coeficientes significativos (es decir, no nulos), y/o para codificar signos de los coeficientes significativos.

En HEVC, la información de posición de los coeficientes de transformada significativos (por ejemplo, el mapa de significación) se codifica en primer lugar para que una TU indique la ubicación del último coeficiente no nulo en el orden de exploración. El mapa de significación y la información de nivel (valores absolutos y los signos de los coeficientes) se codifican para cada coeficiente en un orden de exploración inverso.

Tras producir cualquier transformada coeficientes de transformada, el codificador de vídeo 20 puede realizar la cuantificación de los coeficientes de transformada. La cuantificación se refiere generalmente a un procedimiento en el que se cuantifican los coeficientes de transformada para reducir posiblemente la cantidad de datos usados para representar los coeficientes, proporcionando compresión adicional. El procedimiento de cuantificación puede reducir la profundidad de bit asociada con algunos o todos los coeficientes. Por ejemplo, puede redondearse a la baja un valor de n bits a un valor de m bits durante la cuantificación, donde n es mayor que m . En algunos ejemplos, el codificador de vídeo 20 puede utilizar un orden de exploración predefinido para explorar los coeficientes de transformada cuantificados para producir un vector serializado que puede someterse a codificación de entropía. En otros ejemplos, el codificador de vídeo 20 puede realizar una exploración adaptativa.

Las figuras 2A-2D ilustran algunos órdenes de exploración a modo de ejemplo diferentes. También pueden usarse otros órdenes de exploración definidos, u órdenes de exploración adaptativos (cambiantes). La figura 2A ilustra un orden de exploración en zigzag, la figura 2B ilustra un orden de exploración horizontal, la figura 2C ilustra un orden de exploración vertical y la figura 2D ilustra un orden de exploración diagonal. También pueden definirse y usarse combinaciones de estos órdenes de exploración. En algunos ejemplos, las técnicas de esta divulgación pueden aplicarse específicamente durante la codificación de un denominado mapa de significación en el procedimiento de codificación de vídeo.

Pueden definirse uno o más elementos sintácticos para indicar la posición del último coeficiente significativo (es decir, coeficiente no nulo), que puede depender del orden de exploración asociado con un bloque de coeficientes. Por ejemplo, un elemento sintáctico puede definir una posición de columna del último coeficiente significativo dentro de un bloque de valores de coeficiente y otro elemento sintáctico puede definir una posición de fila del último coeficiente significativo dentro de un bloque de valores de coeficiente.

La figura 3 ilustra un ejemplo de un mapa de significación en relación con un bloque de valores de coeficiente. El mapa de significación se muestra a la derecha, en el que banderas de un bit identifican los coeficientes en el bloque de vídeo a la izquierda que son significativos, es decir, no nulos. En un ejemplo, dados un conjunto de coeficientes significativos (por ejemplo, definidos por un mapa de significación) y un orden de exploración, puede definirse la posición del último coeficiente significativo. En la norma HEVC emergente, los coeficientes de transformada pueden agruparse en un fragmento. El fragmento puede comprender una TU completa o, en algunos casos, las TU pueden subdividirse en fragmentos más pequeños. El mapa de significación y la información de nivel (valor absoluto y signo) se codifican para cada coeficiente en un fragmento. En un ejemplo, un fragmento consiste en 16 coeficientes consecutivos en un orden de exploración inverso (por ejemplo, diagonal, horizontal o vertical) para una TU de 4x4 y una TU de 8x8. Para las TU de 16x16 y 32x32, los coeficientes dentro de un bloque secundario de 4x4 se tratan como un fragmento. Los elementos sintácticos se codifican y señalizan para representar la información de nivel de coeficientes dentro de un fragmento. En un ejemplo, todos los símbolos se codifican en un orden de exploración inverso. Las técnicas de esta divulgación pueden mejorar la codificación de un elemento sintáctico usado para definir esta posición del último coeficiente significativo de un bloque de coeficientes.

Como ejemplo, pueden usarse las técnicas de esta divulgación para codificar la posición del último coeficiente significativo de un bloque de coeficientes (por ejemplo, una TU o un fragmento de una TU). Entonces, después de codificar la posición del último coeficiente significativo, puede codificarse la información de nivel y signo asociada con coeficientes de transformada. La codificación de la información de nivel y signo puede procesarse según un enfoque de cinco pasos codificando los siguientes símbolos en orden de exploración inverso (por ejemplo, para una TU o un fragmento de la TU):

significant_coeff_flag (abrev. como *sigMapFlag*): esta bandera puede indicar la significación de cada coeficiente en un fragmento. Un coeficiente con un valor de uno o mayor puede considerarse significativo.

coeff_abs_level_greater1_flag (abrev. como *gr1Flag*): esta bandera puede indicar si el valor absoluto del coeficiente es mayor de uno para los coeficientes no nulos (es decir, coeficientes con *sigMapFlag* como 1).

coeff_abs_level_greater2_flag (abrev. como *gr2Flag*): esta bandera puede indicar si el valor absoluto del coeficiente es mayor de dos para los coeficientes con un valor absoluto mayor de uno (es decir, coeficientes con *gr1Flag* como 1).

coeff_sign_flag (abrev. como *signFlag*): esta bandera puede indicar la información de signo para los coeficientes no nulos. Por ejemplo, un cero para esta bandera puede indicar un signo positivo, mientras que un 1 puede indicar un signo negativo.

coeff_abs_level_remain (abrev. como *levelRem*): este elemento sintáctico puede indicar el valor absoluto restante de un nivel de coeficiente de transformada. Para este elemento sintáctico, el valor absoluto del coeficiente - x puede codificarse como $(\text{abs}(\text{nivel}) - x)$ para cada coeficiente con la amplitud mayor de x. El valor de x depende de la presencia de *gr1Flag* y *gr2Flag*. Si se ha codificado *gr2Flag*, el valor de *levelRem* se calcula como $(\text{abs}(\text{nivel}) - 2)$. Si no se ha codificado *gr2Flag*, pero se ha codificado *gr1Flag*, el valor de *levelRem* se calcula como $(\text{abs}(\text{nivel}) - 1)$.

De esta manera, pueden codificarse coeficientes de transformada para una TU o una parte (por ejemplo, fragmento) de una TU. En cualquier caso, las técnicas de esta divulgación, que se refieren a la codificación de un elemento sintáctico usado para definir la posición del último coeficiente significativo de un bloque de coeficientes, también pueden usarse con otros tipos de técnicas para codificar, en última instancia, la información de nivel y signo de los coeficientes de transformada. El enfoque de cinco pasos para codificar información de nivel, signo y significación es solamente una técnica a modo de ejemplo que puede usarse tras la codificación de la posición del último coeficiente significativo de un bloque, tal como se expone en esta divulgación.

Después de explorar los coeficientes de transformada cuantificados para formar un vector unidimensional, el codificador de vídeo 20 puede someter a codificación de entropía el vector unidimensional de los coeficientes de transformada. El codificador de vídeo 20 también puede someter a codificación de entropía elementos sintácticos asociados con los datos de vídeo codificados para su uso por el decodificador de vídeo 30 en la decodificación de

los datos de vídeo. Puede realizarse codificación de entropía según una de las técnicas siguientes: CAVLC, CABAC, codificación aritmética binaria adaptativa según el contexto basada en sintaxis (SBAC), codificación PIPE u otra metodología de codificación de entropía. Para realizar CAVLC, el codificador de vídeo 20 puede seleccionar un código de longitud variable para un símbolo que va a transmitirse. Pueden construirse palabras de código en la codificación de longitud variable (VLC) de tal manera que códigos relativamente más cortos correspondan a símbolos más probables, mientras que códigos relativamente más largos correspondan a símbolos menos probables. De esta manera, el uso de VLC puede lograr ahorros de bits con respecto a usar, por ejemplo, palabras de código de igual longitud para cada símbolo que va a transmitirse.

Las técnicas de codificación de entropía de esta divulgación se describen específicamente como que son aplicables a CABAC, aunque las técnicas también pueden ser aplicables a CAVLC, SBAC, PIPE, u otras técnicas. Obsérvese que PIPE usa principios similares a los de la codificación aritmética.

En general, los símbolos de datos de codificación que usan CABAC pueden implicar una o más de las siguientes etapas:

(1) Binarización: Si un símbolo que va a codificarse no tiene un valor binario, se mapea en una secuencia binaria, es decir, una denominada "cadena de bins". Cada índice binario (es decir, "bin") en la cadena de bins puede tener un valor de "0" o "1".

(2) Asignación de contexto: En el modo normal, se asigna cada bin a un contexto. También puede codificarse un bin según un modo de derivación en el que no se asigna un contexto. Un contexto es un modelo de probabilidad y se denomina a menudo "modelo de contexto" Tal como se usa en el presente documento, el término contexto puede referirse a un modelo de probabilidad o un valor de probabilidad. Un contexto determina cómo se calcula la probabilidad del valor de un bin para un bin dado. El contexto puede asociar la probabilidad del valor de un bin basándose en información tal como valores de símbolos codificados previamente o un número de bins. Además, puede asignarse un contexto a un bin basándose en parámetros de nivel (por ejemplo, segmento) superior.

(3) Codificación de bins: Los bins se codifican con un codificador aritmético. Para codificar un bin, el codificador aritmético requiere como entrada la probabilidad del valor del bin, es decir, la probabilidad de que el valor del bin sea igual a "0", y/o la probabilidad de que el valor del bin sea igual a "1". La probabilidad (estimada) puede representarse en un contexto por un valor entero denominado "estado de contexto"

(4) Actualización de estado: La probabilidad (estado) para un contexto seleccionado puede actualizarse basándose en el valor codificado real del bin (por ejemplo, si el valor de bin era de "1", puede aumentarse la probabilidad de que un bin sea "1"). La actualización de probabilidades puede estar dictada según las reglas de transición asociadas con un contexto.

Lo siguiente es un una técnica de binarización a modo de ejemplo de los elementos sintácticos del último coeficiente significativo que puede realizarse mediante el codificador de vídeo 20. Los elementos sintácticos del último coeficiente significativo pueden incluir una posición de fila y columna del último coeficiente significativo dentro de un bloque bidimensional (es decir, las coordenadas x e y). Para un bloque de 8×8 , existen ocho posibilidades diferentes para la última posición de un coeficiente dentro de una columna o fila, es decir, $0, 1, \dots, 7$. Se usan ocho bins diferentes para representar estas ocho posiciones de fila o columna. Por ejemplo, $\text{bin}_0 = 1$ puede indicar que el coeficiente en la fila o columna 0 es el último coeficiente significativo. En este ejemplo, si $\text{bin}_0 = 0$, entonces el coeficiente en la ubicación 0 no es el último coeficiente. Otro bin igual a 1 puede indicar la posición del último coeficiente significativo. Por ejemplo, $\text{bin}_1 = 1$ puede indicar que la ubicación 1 es el último coeficiente significativo. Como otro ejemplo, $\text{bin}_X = 1$ puede indicar que la ubicación X es el último coeficiente significativo. Tal como se describió anteriormente, cada bin puede codificarse mediante dos métodos diferentes: (1) codificar el bin con un contexto y (2) codificar el bin usando el modo de derivación (sin un contexto).

La tabla 1 muestra una binarización a modo de ejemplo de la posición del último coeficiente significativo en la que algunos bins se codifican con un contexto y otros se codifican usando un modo de derivación. El ejemplo en la tabla 1 proporciona una binarización a modo de ejemplo del último coeficiente significativo para una TU de 32×32 . La segunda columna de la tabla 1 proporciona valores de prefijo unitario truncado correspondientes para valores posibles de la posición del último coeficiente significativo dentro de una TU de tamaño T definido por la longitud de prefijo unitario truncado máxima de $2 \log_2(T) - 1$. La tercera columna de la tabla 1 proporciona un sufijo de longitud fija correspondiente para cada prefijo unitario truncado. Por motivos de brevedad, la tabla 1 incluye valores de X que indican un valor de bit o bien de uno o bien de cero. Obsérvese que los valores de X mapean de manera única cada valor que comparte un prefijo unitario truncado según un código de longitud fija. La magnitud de la componente de última posición en la tabla 1 puede corresponder a un valor de coordenada x y/o un valor de coordenada y . Obsérvese que la binarización del último coeficiente significativo para una TU de 4×4 , 8×8 y 16×16 puede definirse de manera similar a la binarización de una TU de 32×32 descrita con respecto a la tabla 1.

Magnitud del componente de última posición	(Modelo de contexto) unitario truncado	(Derivación) binaria fija	valor de <i>f</i> (<i>f_value</i>)
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

Tabla 1. Binarización para una TU de tamaño 32x32, donde X significa 1 o 0.

Tal como se describió anteriormente, los símbolos de datos de codificación que usan CABAC también pueden implicar asignación de contexto. En un ejemplo, puede usarse modelado de contexto para la codificación aritmética de la parte de cadenas unitarias fijas de la cadena de bins, mientras que no se usa modelado de contexto para la codificación aritmética de las cadenas binarias fijas de la cadena de bins (es decir, se somete a derivación la cadena binaria fija). En el caso en el que se codifican cadenas unitarias truncadas usando modelado de contexto, puede asignarse un contexto a cada índice de bin de una cadena binaria.

Existen varias maneras en las que pueden asignarse contextos a cada índice de bin de una cadena binaria. El número de asignaciones de contexto para una cadena de bins que representa la posición del último coeficiente significativo puede ser igual al número de índices de bin o longitud de una cadena unitaria truncada para posibles tamaños de TU y componentes de color. Por ejemplo, si los posibles tamaños de una componente de luma son 4x4, 8x8, 16x16 y 32x32, el número de asignaciones de contexto para una dimensión puede ser igual a 60 (es decir, 4+8+16+32), cuando no se somete a derivación ningún bin. Asimismo, para cada componente de croma con tamaños posibles de 4x4, 8x8 y 16x16, el número de asignaciones de contexto puede ser igual a 28 (es decir, 4+8+16), cuando no se somete a derivación ninguno de los bins. Por tanto, el número máximo de asignaciones de contexto puede ser igual a 116 (es decir, 60+28+28) para cada dimensión cuando se especifica la posición del último coeficiente significativo usando coordenadas tanto x como y. Las asignaciones de contexto a modo de ejemplo a continuación suponen que algunos bins se someterán a derivación según el esquema de binarización descrito con respecto a la tabla 1. Sin embargo, obsérvese que las técnicas de asignación de contexto descritas en el presente documento pueden ser aplicables a varios esquemas de binarización. Además, incluso si se supone se someterán a derivación algunos bins, existen todavía numerosas maneras en las que pueden asignarse contextos a bins de una cadena de bins que representa la posición del último coeficiente significativo.

En algunos casos, puede ser deseable reducir el número total de contextos en relación con el número requerido de asignaciones de contexto. De esta manera, puede no ser necesario que un codificador o decodificador almacene y mantenga tantos contextos. Sin embargo, cuando se reduce el número de contextos, también puede reducirse la precisión de predicción, por ejemplo, si se comparten contextos por dos bins con probabilidades diferentes. Además, puede actualizarse un contexto particular con mayor frecuencia, lo que puede afectar a la probabilidad estimada del valor de un bin. Es decir, codificar un bin usando un contexto asignado puede implicar actualizar un contexto. Por tanto, un bin posterior asignado al contexto puede codificarse usando el contexto actualizado. Además, obsérvese que en algunos ejemplos pueden inicializarse modelos de contexto a nivel de segmento, de tal manera que los valores de los bins dentro de un segmento pueden no afectar a la codificación de bins dentro de un segmento posterior, aunque se les asigne a los bins el mismo contexto. Esta divulgación describe técnicas para optimizar asignaciones de contexto, de tal manera que pueda reducirse el número de contextos al tiempo que se mantiene la precisión para las probabilidades estimadas. En un ejemplo, las técnicas de asignación de contexto descritas en el presente documento incluyen técnicas en las que se les asigna el mismo contexto a índices de bin individuales.

Las tablas 2-13 a continuación ilustran asignaciones de contexto para los índices de bin de una cadena de bins que representa la posición del último coeficiente significativo dentro de una TU. Obsérvese que para algunos bins en las tablas 2-13 (por ejemplo, los bins 5-7 de un bloque de 8x8) no existe ningún contexto asignado. Esto es debido a que se supone que estos bins se codificarán usando un modo de derivación, tal como se describió anteriormente. Obsérvese también que los valores en la tabla 2-13 representan un índice de contexto. En las tablas 2-13, cuando bins diferentes tienen el mismo valor de índice de contexto, comparten el mismo contexto. El mapeo de un índice de contexto en un contexto real puede definirse según una norma de codificación de vídeo. Las tablas 2-13 ilustran cómo puede asignarse habitualmente un contexto a bins.

La tabla 2 ilustra una posible indexación de contexto para cada bin de tamaños de TU diferentes para las binarizaciones a modo de ejemplo proporcionadas anteriormente con respecto a la tabla 1 anterior. En el ejemplo en la tabla 2, se permite que bins vecinos compartan los mismos contextos. Por ejemplo, los bins 2 y 3 de una TU de 8x8 comparten el mismo contexto.

ES 2 700 523 T3

Índice de bin	0	1	2	3	4	5	6	7	8
TU de 4x4	0	1	2						
TU de 8x8	3	4	5	5	6				
TU de 16x16	7	8	9	9	10	10	11		
TU de 32x32	12	13	14	14	15	15	16	16	17

Tabla 2: Asignación de contexto para la codificación de la última posición

Las tablas 3-6 ilustran, cada una, ejemplos adicionales de asignaciones de contexto según las reglas siguientes:

1. Los primeros bins K no comparten contextos, donde $K > 1$. K podría ser diferente para cada tamaño de TU.
2. Un contexto solo puede asignarse a bins consecutivos. Por ejemplo, los bins 3 a 5 podrían usar el contexto 5. Sin embargo, no está permitido que el bin 3 y el bin 5 usen el contexto 5 y el bin 4 use el contexto 6.
3. Los últimos bin N , $N \geq 0$, de tamaños de TU diferentes pueden compartir el mismo contexto.
4. El número de bins que comparten el mismo contexto aumenta con los tamaños de TU.

Las reglas 1-4 anteriores pueden ser particularmente útiles para la binarización proporcionada en la tabla 1. Sin embargo, las reglas 1-4 pueden ser igualmente útiles para otros esquemas de binarización y las asignaciones de contexto reales pueden ajustarse, en consecuencia, al esquema de binarización que se implementa.

Índice de bin	0	1	2	3	4	5	6	7	8
TU de 4x4	0	1	2						
TU de 8x8	3	4	5	6	7				
TU de 16x16	8	9	10	11	11	12	12		
TU de 32x32	13	14	14	15	16	16	16	16	17

Tabla 3: Ejemplo de bins de la última posición según las reglas 1-4

Índice de bin	0	1	2	3	4	5	6	7	8
TU de 4x4	0	1	2						
TU de 8x8	3	4	5	6	6				
TU de 16x16	8	9	10	11	11	12	12		
TU de 32x32	13	14	14	15	16	16	16	16	17

Tabla 4: Ejemplo de bins de la última posición según las reglas 1-4

Índice de bin	0	1	2	3	4	5	6	7	8
TU de 4x4	0	1	2						
TU de 8x8	3	4	5	6	7				
TU de 16x16	8	9	10	11	11	12	12		
TU de 32x32	13	14	14	15	16	16	16	12	12

Tabla 5: Ejemplo de bins de la última posición según las reglas 1-4

Índice de bin	0	1	2	3	4	5	6	7	8
TU de 4x4	0	1	2						
TU de 8x8	3	4	5	6	7				
TU de 16x16	8	9	10	10	11	11	12		
TU de 32x32	13	14	14	15	15	15	16	16	16

Tabla 6: Ejemplo de bins de la última posición según las reglas 1-4

- Las tablas 7-8 a continuación proporcionan asignaciones de contexto a modo de ejemplo en el que los últimos bins de tamaños de bloque diferentes comparten el mismo contexto, lo que puede optimizar adicionalmente el número de contextos. En un ejemplo, puede usarse mapeo directo para determinar cómo se comparten los contextos entre los últimos bins de dos o más tamaños de bloque. Por ejemplo, para un bloque A y un bloque B con tamaños M y N , respectivamente, el contexto del bin de orden n del bloque A puede usar el mismo contexto que el bin de orden n de tamaño de bloque B.

Índice de bin	0	1	2	3	4	5	6	7	8
TU de 4x4	12	13	14						
TU de 8x8	12	13	14	14	15				
TU de 16x16	12	13	14	14	15	15	16		
TU de 32x32	12	13	14	14	15	15	16	16	17

Tabla 7: Ejemplo de bins de la última posición en los que tamaños de bloque comparten los mismos contextos.

La tabla 8 muestra otro ejemplo en el que los bins de la última posición de algunos tamaños de bloque comparten contextos entre sí. En este caso, TU de tamaño 8x8 y 16x16 comparten los mismos contextos.

Índice de bin	0	1	2	3	4	5	6	7	8
TU de 4x4	0	1	2						
TU de 8x8	7	8	9	9	10				
TU de 16x16	7	8	9	9	10	10	11		
TU de 36x36	12	13	14	14	15	15	16	16	17

Tabla 8: Ejemplo de bins de la última posición en los que algunos tamaños de bloque (8x8 y 16x16) comparten los mismos contextos.

5 En otro ejemplo, el mapeo de contextos para bins de la última posición de tamaños de bloque diferentes puede derivarse usando una función $f(.)$. Por ejemplo, el bin de orden n en el tamaño de bloque A puede compartir los mismos contextos con el bin de orden m en el tamaño de bloque B , donde m es una función de n ($m = f(n)$). Por ejemplo, la función puede ser lineal, es decir, $m = n \cdot a + b$, donde a y b son parámetros de la función lineal. La tabla 9 muestra un ejemplo en el que $a = 1$, $b = 1$, $A =$ una TU de 8x8 y $B =$ una TU de 16x16.

Índice de bin	0	1	2	3	4	5	6	7	8
TU de 4x4	0	1	2						
TU de 8x8	8	9	9	10	10				
TU de 16x16	7	8	9	9	10	10	11		
TU de 36x36	12	13	14	14	15	15	16	16	17

Tabla 9: Ejemplo de bins de la última posición con contextos compartidos basándose en una función lineal

10 Obsérvese que cuando se aplica la ecuación anterior en determinados casos, debido a la operación entera, puede haber redondeo implicado. Por ejemplo, $7 \cdot 0,5 = 3$.

Según el siguiente ejemplo, el mapeo desde la ubicación n en un tamaño de bloque de 8x8 a una ubicación m en un bloque de 4x4 puede calcularse con la siguiente ecuación:

$$m = f(n) = n \gg 1, \text{ lo que significa que } a = 0,5, b = 0, A = 8 \times 8, B = 4 \times 4 \quad (1)$$

15 El mapeo desde la ubicación n en un bloque de 16x16 hasta una ubicación m en un bloque de 4x4 puede calcularse con la siguiente ecuación:

$$m = f(n) = n \gg 2, \text{ lo que significa que } a = 0,5, b = 0, A = 16 \times 16, B = 4 \times 4 \quad (2)$$

20 Tal como se describió anteriormente, las ecuaciones (1) y (2) son solamente un par de ejemplos que pueden usarse para implementar un mapeo entre bloques de tamaños diferentes. Las ecuaciones (1) y (2) pueden denominarse funciones de mapeo. Obsérvese que el " \gg " en las ecuaciones (1) y (2) puede representar una operación de desplazamiento definida según una norma de codificación de vídeo, tal como HEVC. Además, pueden usarse otras ecuaciones para lograr el mismo mapeo o mapeos diferentes.

La tabla 10 proporciona una asignación de contexto a modo de ejemplo para el último coeficiente significativo para TU de 4x4, 8x8 y 16x16 según las ecuaciones (1) y (2).

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

Tabla 10: Mapeo de contexto para unidades de transformada de tamaño diferente

25 La tabla 11 proporciona un ejemplo de la asignación de contexto de la tabla 10 en la que se usan valores de índice de contexto diferentes (es decir, 15-17 en lugar de 0-2). Tal como se describió anteriormente, no se pretende que los valores de los índices de contexto en las tablas 3-12 limiten los contextos reales asignados a un índice binario.

Índice de bin	0	1	2	3	4	5	6	7	8
TU de 4x4	15	16	17						
TU de 8x8	15	15	16	16	17				
TU de 16x16	15	15	15	15	16	16	16		

Tabla 11: Mapeo de contexto para unidades de transformada de tamaño diferente

Obsérvese que el mapeo de los contextos en la tabla 11 es equivalente a la siguiente función de mapeo:

$$30 \text{ ctx_index} = (n \gg k) + 15 \quad (3)$$

donde ctx_index es el índice del contexto;

$n =$ índice de bin

$k = \log_2 \text{TrafoDimension} - 2;$

$\log_2\text{TrafoDimension} = \log_2(\text{anchura})$ para la última posición en la dimensión x;

$\log_2\text{TrafoDimensión} = \log_2(\text{altura})$ para la última posición en la dimensión y.

En algunos casos, la función definida en (1)-(3) puede usarse por un dispositivo de codificación para construir una serie de tablas que pueden almacenarse en la memoria y utilizarse para consultar asignaciones de contexto. En algunos casos, las tablas pueden determinarse basándose en las ecuaciones y reglas descritas en el presente documento y almacenarse tanto en el codificador de vídeo 20 como en el decodificador de vídeo 30.

Además, en algunos ejemplos, las funciones (1)-(3) definidas anteriormente pueden aplicarse de manera selectiva para asignar contextos para bins particulares. De esta manera, se les puede asignar un contexto a bins diferentes basándose en reglas diferentes. En un ejemplo, las funciones, tales como las descritas anteriormente, pueden ser aplicables únicamente para un índice de bin (es decir, un valor de n) que es menor que un umbral Th1, y/o mayor que un umbral Th2. La tabla 12 muestra un ejemplo en el que las técnicas de mapeo descritas anteriormente se aplican de manera selectiva basándose en el valor del índice de bin, es decir, $n > \text{Th}2 = 2$.

Índice de bin	0	1	2	3	4	5	6	7	8
TU de 4x4	0	1	2						
TU de 8x8	3	4	9	10	10				
TU de 16x16	7	8	9	9	10	10	11		
TU de 32x32	12	13	14	14	15	15	16	16	17

Tabla 12: Ejemplo de bins de la última posición con contextos compartidos basándose en una función lineal y un umbral

En otro ejemplo, el valor de umbral para aplicar técnicas a índices de bin puede ser diferente para tamaños de bloque diferentes, tipos de fotograma diferentes, componentes de color diferentes (Y,U,V), y/u otra información adicional. Este umbral puede predefinirse según una norma de codificación de vídeo o puede señalizarse usando sintaxis de alto nivel. Por ejemplo, el umbral puede señalizarse en un conjunto de parámetros de secuencia (SPS), un conjunto de parámetros de imagen (PPS), un conjunto de parámetros de adaptación (APS) y/o una cabecera de segmento.

En otro ejemplo, una función de mapeo puede ser diferente para tamaños de bloque diferentes, tipos de fotograma diferentes, componentes de color diferentes (Y, U y V), y/u otra información adicional. La función de mapeo puede predefinirse según una norma de codificación de vídeo o puede señalizarse usando sintaxis de alto nivel. Por ejemplo, la función de mapeo puede señalizarse en un SPS, un PPS, un APS y/o una cabecera de segmento.

En otro ejemplo, las técnicas de mapeo directo y mapeo de función descritas anteriormente pueden aplicarse de manera adaptativa basándose en componentes de color, tipo de fotograma, parámetro de cuantificación (QP) y/u otra información adicional. Por ejemplo, las técnicas de mapeo directo o mapeo de función pueden aplicarse solamente a las componentes de croma. Las reglas para esta capacidad adaptativa pueden predefinirse o pueden señalizarse usando sintaxis de alto nivel. Por ejemplo, las reglas para la capacidad adaptativa pueden señalizarse en un SPS, un PPS, un APS y/o una cabecera de segmento.

En otro ejemplo, bins de la última posición para componentes de croma y luma pueden compartir el mismo contexto. Esto puede ser aplicable para cualquier tamaño de bloque, por ejemplo, 4x4, 8x8, 16x16 ó 32x32. La tabla 13 muestra un ejemplo en el que se comparten contextos por bins de la última posición componentes de luma y de croma para una TU de 4x4.

Índice de bin	0	1	2	3
Luma, TU de 4x4	0	1	2	
Croma, TU de 4x4	0	1	2	

Tabla 13: Ejemplo de bins de la última posición para componentes de croma y luma en una TU de 4x4

La figura 4 es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo 20 que puede usar técnicas de codificación de coeficientes de transformada según se describe en esta divulgación. Por ejemplo, el codificador de vídeo 20 representa un ejemplo de un codificador de vídeo configurado para obtener una cadena binaria que indica la posición del último coeficiente significativo dentro de un bloque de vídeo; determinar un contexto para un índice binario de la cadena binaria basándose en el tamaño de bloque de vídeo, en el que el contexto se asigna a al menos dos índices binarios, en el que cada uno de los al menos dos índices binarios se asocian con tamaños de bloque de vídeo diferentes; y codificar la cadena binaria usando CABAC basada al menos en parte en el contexto determinado. El codificador de vídeo 20 se describirá en el contexto de la codificación de HEVC con propósitos ilustrativos, pero sin limitación de esta divulgación con respecto a otras normas o métodos de codificación que puedan requerir la exploración de coeficientes de transformada. El codificador de vídeo 20 puede realizar codificación intra e inter de CU dentro de fotogramas de vídeo. La codificación intra se basa en la predicción espacial para reducir o eliminar redundancia espacial en datos de vídeo dentro de un fotograma de vídeo dado. La codificación inter se basa en la predicción temporal para reducir o eliminar redundancia temporal entre un fotograma actual y fotogramas codificados previamente de una secuencia de vídeo. El modo intra (modo I) puede referirse a

cualquiera de varios modos de compresión de vídeo de base espacial. Los modos inter tales como la predicción unidireccional (modo P) o predicción bidireccional (modo B) pueden referirse a cualquiera de varios modos de compresión de vídeo de base temporal.

5 Tal como se muestra en la figura 4, el codificador de vídeo 20 recibe un bloque de vídeo actual dentro de un fotograma de vídeo que va a codificarse. En el ejemplo de la figura 4, el codificador de vídeo 20 incluye un módulo de selección de modo 40, un módulo de estimación de movimiento 42, un módulo de compensación de movimiento 44, un módulo de predicción intra 46, una memoria intermedia de fotograma de referencia 64, un sumador 50, un módulo de transformada 52, un módulo de cuantificación 54 y un módulo de codificación de entropía 56. El módulo de transformada 52 ilustrado en la figura 4 es el módulo que aplica la transformada real o combinaciones de transformada a un bloque de datos residuales, y no debe confundirse con un bloque de coeficientes de transformada, que también pueden denominarse una unidad de transformada (TU) de una CU. Para una reconstrucción de bloque de vídeo, el codificador de vídeo 20 también incluye un módulo de cuantificación inversa 58, un módulo de transformada inversa 60 y un sumador 62. Un filtro de desbloqueo (no mostrado en la figura 4) puede incluirse también para filtrar límites de bloque para eliminar artefactos con capacidad de bloqueo del vídeo reconstruido. Si se desea, el filtro de desbloqueo filtrará normalmente la salida del sumador 62.

20 Durante el procedimiento de codificación, el codificador de vídeo 20 recibe un segmento o fotograma de vídeo que va a codificarse. El fotograma o segmento puede dividirse en múltiples bloques de vídeo, por ejemplo, unidades de codificación de mayor tamaño (LCU). El módulo de estimación de movimiento 42 y el módulo de compensación de movimiento 44 realizan codificación de predicción inter del bloque de vídeo recibido en relación con uno o más bloques en uno o más fotogramas de referencia para proporcionar compresión temporal. El módulo de predicción intra 46 puede realizar codificación de predicción intra del bloque de vídeo recibido en relación con uno o más bloques vecinos en el mismo fotograma o segmento como el bloque que va a codificarse para proporcionar compresión espacial.

25 El módulo de selección de modo 40 puede seleccionar uno de los modos de codificación, intra o inter, por ejemplo, basándose en resultados de error (es decir, de distorsión) para cada modo, y proporciona el bloque de predicción intra o inter resultante (por ejemplo, una unidad de predicción (PU)) al sumador 50 para generar datos de bloques residuales y al sumador 62 para reconstruir el bloque codificado para su uso en un fotograma de referencia. El sumador 62 combina el bloque predicho con datos sometidos a transformación inversa y sometidos a cuantificación inversa procedentes del módulo de transformada inversa 60 para que el bloque reconstruya el bloque codificado, según se describe en mayor detalle a continuación. Algunos fotogramas de vídeo pueden designarse como fotogramas I, en los que todos los bloques en un fotograma I se codifican en un modo de predicción intra. En algunos casos, el módulo de predicción intra 46 puede realizar codificación de predicción intra de un bloque en un fotograma P o B, por ejemplo, cuando la búsqueda de movimiento realizada por el módulo de estimación de movimiento 42 no da como resultado una predicción suficiente del bloque.

35 El módulo de estimación de movimiento 42 y el módulo de compensación de movimiento 44 pueden estar altamente integrados, pero se ilustran de manera independiente con propósitos conceptuales. La estimación de movimiento (o búsqueda de movimiento) es el procedimiento de generación de vectores de movimiento, que estiman movimiento para bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de una unidad de predicción en un fotograma actual en relación con una muestra de referencia de un fotograma de referencia. El módulo de estimación de movimiento 42 calcula un vector de movimiento para una unidad de predicción de un fotograma sometido a codificación intra comparando la unidad de predicción con muestras de referencia de un fotograma de referencia almacenado en la memoria intermedia de fotograma de referencia 64. Una muestra de referencia puede ser un bloque que se encuentra que coincide en gran medida con la parte de la CU que incluye la PU que está codificándose, en términos de diferencia de píxel, lo que puede determinarse mediante la suma de diferencias absolutas (SAD), suma de diferencias al cuadrado (SSD), o u otras métrica de diferencia. La muestra de referencia puede aparecer en cualquier lugar dentro de un fotograma de referencia o segmento de referencia, y no necesariamente en un límite de bloque (por ejemplo, unidad de codificación) del segmento o fotograma de referencia . En algunos ejemplos, la muestra de referencia puede aparecer en una posición de píxel fraccionaria.

50 El módulo de estimación de movimiento 42 envía el vector de movimiento calculado al módulo de codificación de entropía 56 y al módulo de compensación de movimiento 44. La parte del fotograma de referencia identificada por un vector de movimiento puede denominarse muestra de referencia. El módulo de compensación de movimiento 44 puede calcular un valor de predicción para una unidad de predicción de una CU actual, por ejemplo, recuperando la muestra de referencia identificada por un vector de movimiento para la PU.

55 El módulo de predicción intra 46 puede someter a predicción intra el bloque recibido, como alternativa a la predicción inter realizada por el módulo de estimación de movimiento 42 y el módulo de compensación de movimiento 44. El módulo de predicción intra 46 puede predecir el bloque recibido en relación con bloques vecinos codificados previamente, por ejemplo, bloques por encima, por encima y a la derecha, por encima y a la izquierda, o a la izquierda del bloque actual, suponiendo un orden de codificación de izquierda a derecha y de arriba a abajo para los bloques. El módulo de predicción intra 46 puede configurarse con diversos modos de predicción intra diferentes. Por ejemplo, el módulo de predicción intra 46 puede configurarse con un número determinado de modos de predicción direccionales, por ejemplo, treinta y cuatro modos de predicción direccionales, basándose en el tamaño de la CU

que está codificándose.

El módulo de predicción intra 46 puede seleccionar un modo de predicción intra, por ejemplo, calculando valores de error para diversos modos de predicción intra y seleccionando un modo que produzca el valor de error más bajo. Los modos de predicción direccionales pueden incluir funciones para combinar valores de píxeles vecinos espacialmente y aplicar los valores combinados a una o más posiciones de píxel en una PU. Una vez que se han calculado los valores para todas las posiciones de píxel en la PU, el módulo de predicción intra 46 puede calcular un valor de error para el modo de predicción basándose en diferencias de píxel entre la PU y el bloque recibido que va a codificarse. El módulo de predicción intra 46 puede seguir sometiendo a prueba modos de predicción intra hasta que se descubra un modo de predicción intra que produzca un valor de error aceptable. El módulo de predicción intra 46 puede enviar entonces la PU al sumador 50.

El codificador de vídeo 20 forma un bloque residual restando los datos de predicción calculados por el módulo de compensación de movimiento 44 o el módulo de predicción intra 46 a partir del bloque de vídeo original que está codificándose. El sumador 50 representa la componente o componentes que realizan esta operación de resta. El bloque residual puede corresponder a una matriz bidimensional de valores de diferencia de píxel, en la que el número de valores en el bloque residual es el mismo que el número de píxeles en la PU correspondiente al bloque residual. Los valores en el bloque residual pueden corresponder a la diferencias, es decir, error, entre valores de píxeles ubicados conjuntamente en la PU y en el bloque original que va a codificarse. La diferencias pueden ser diferencias de croma o luma dependiendo del tipo de bloque que se codifica.

El módulo de transformada 52 puede formar una o más unidades de transformada (TU) a partir del bloque residual. El módulo de transformada 52 selecciona una transformada de una pluralidad de transformadas. La transformada puede seleccionarse basándose en una o más características de codificación, tales como tamaño de bloque, modo de codificación, o similar. El módulo de transformada 52 aplica entonces la transformada seleccionada a la TU, produciendo un bloque de vídeo que comprende una matriz bidimensional de coeficientes de transformada.

El módulo de transformada 52 puede enviar los coeficientes de transformada resultantes al módulo de cuantificación 54. El módulo de cuantificación 54 puede cuantificar entonces los coeficientes de transformada. El módulo de codificación de entropía 56 puede realizar entonces una exploración de los coeficientes de transformada cuantificados en la matriz según un modo de exploración. Esta divulgación describe que el módulo de codificación de entropía 56 realiza la exploración. Sin embargo, debe entenderse que, en otros ejemplos, otros módulos de procesamiento, tales como el módulo de cuantificación 54, podrían realizar la exploración.

El módulo de cuantificación inversa 58 y el módulo de transformada inversa 60 aplican cuantificación inversa y transformación inversa, respectivamente, para reconstruir el bloque residual en el dominio de píxeles, por ejemplo, para usarlo después como bloque de referencia. El módulo de compensación de movimiento 44 puede calcular un bloque de referencia añadiendo el bloque residual a un bloque predictivo de uno de los fotogramas de la memoria intermedia de fotograma de referencia 64. La memoria intermedia de fotograma de referencia 64 se denomina a veces memoria intermedia de imagen decodificada (DPB). El módulo de compensación de movimiento 44 puede aplicar también uno o más filtros de interpolación al bloque residual reconstruido para calcular valores de píxel enteros secundarios para su uso en la estimación de movimiento. El sumador 62 añade el bloque residual reconstruido al bloque de predicción con movimiento compensado producido por el módulo de compensación de movimiento 44 para producir un bloque de vídeo reconstruido para su almacenamiento en la memoria intermedia de fotograma de referencia 64. El bloque de vídeo reconstruido puede usarse por el módulo de estimación de movimiento 42 y el módulo de compensación de movimiento 44 como bloque de referencia para someter a codificación inter un bloque en un fotograma de vídeo posterior.

Una vez que los coeficientes de transformada se someten a exploración en la matriz unidimensional, el módulo de codificación de entropía 56 puede aplicar codificación de entropía tal como CAVLC, CABAC, SBAC, PIPE, u otra metodología de codificación de entropía a los coeficientes. En algunos casos, el módulo de codificación de entropía 56 puede configurarse para realizar otras funciones de codificación, además de codificación de entropía. Por ejemplo, el módulo de codificación de entropía 56 puede configurarse para determinar valores de patrón de bloque codificado (CBP) para CU y PU. Además, en algunos casos, el módulo de codificación de entropía 56 puede realizar codificación por longitud de recorrido de coeficientes. Tras la codificación de entropía por el módulo de codificación de entropía 56, el vídeo codificado resultante puede transmitirse a otro dispositivo, tal como el decodificador de vídeo 30, o archivarse para una transmisión o recuperación posterior.

Según las técnicas de esta divulgación, el módulo de codificación de entropía 56 puede seleccionar el contexto usado para codificar elementos sintácticos basándose, por ejemplo, en las asignaciones de contexto descritas anteriormente con respecto a las tablas 2-13 y cualquier combinación de los siguientes: una dirección de predicción intra para modos de predicción intra, una posición de exploración del coeficiente correspondiente a los elementos sintácticos, tipo de bloque, tipo de transformada, y/u otras propiedades de secuencia de vídeo.

En un ejemplo, el módulo de codificación de entropía 56 puede codificar la posición del último coeficiente significativo usando la técnica de binarización adoptada en HEVC descrita anteriormente con respecto a la tabla 1. En otros ejemplos, el módulo de codificación de entropía 56 puede codificar la posición del último coeficiente

significativo usando otras técnicas de binarización. En un ejemplo, una palabra de código para la posición del último coeficiente significativo puede incluir un prefijo de código unitario truncado seguido por un código de sufijo de longitud fija. En un ejemplo, cada magnitud de última posición puede usar la misma binarización para todos los tamaños posibles de TU, salvo cuando la última posición es igual al tamaño de TU menos 1. Esta excepción se debe a las propiedades de la codificación unitaria truncada. En un ejemplo, puede especificarse la posición del último coeficiente significativo dentro de un coeficiente de transformada rectangular especificando un valor de coordenada x y un valor de coordenada y. En otro ejemplo, un bloque de coeficiente de transformada puede estar en forma de un vector $1 \times N$ y puede especificarse la posición del último coeficiente significativo dentro del vector mediante un único valor de posición.

5 La figura 5 es un diagrama de bloques que ilustra un módulo de codificación de entropía 56 a modo de ejemplo que puede implementar las técnicas descritas en esta divulgación. En un ejemplo, el módulo de codificación de entropía 56 ilustrado en la figura 5 puede ser un codificador CABAC. El módulo de codificación de entropía 56 a modo de ejemplo puede incluir un módulo de binarización 502, un módulo de codificación aritmética 510, que incluye un motor de codificación de derivación 504 y un motor de codificación normal 508, y un módulo de modelado de contexto 506.

10 El módulo de codificación de entropía 56 recibe elementos sintácticos, tales como uno o más elementos sintácticos que representan la posición del último coeficiente de transformada significativo dentro de un bloque de coeficientes de transformada y codifica el elemento sintáctico en un flujo de bits. Los elementos sintácticos pueden incluir un elemento sintáctico que especifica una coordenada x de la posición del último coeficiente significativo dentro de un bloque de coeficiente de transformada y un elemento sintáctico que especifica una coordenada y de la posición del último coeficiente significativo dentro de un bloque de coeficiente de transformada.

15

20

El módulo de binarización 502 recibe un elemento sintáctico y produce una cadena de bins (es decir, cadena binaria). En un ejemplo, el módulo de binarización 502 recibe elementos sintácticos que representan la última posición de un coeficiente significativo dentro de un bloque de coeficientes de transformada y produce una cadena de bins según el ejemplo descrito anteriormente con respecto a la tabla 1. El módulo de codificación aritmética 510 recibe una cadena de bins procedente del módulo de binarización 502 y realiza codificación aritmética en la cadena de bins. Tal como se muestra en la figura 5, el módulo de codificación aritmética 510 puede recibir valores de bin procedentes de una ruta de derivación o la ruta de codificación normal. De manera consistente con el procedimiento CABAC descrito anteriormente, en el caso en el que el módulo de codificación aritmética 510 recibe valores de bin procedentes de una ruta de derivación, el motor de codificación de derivación 504 puede realizar codificación aritmética con valores de bin sin utilizar un contexto asignado a un valor de bin. En un ejemplo, el motor de codificación de derivación 504 puede suponer probabilidades iguales para valores posibles de un bin.

25

30

En el caso en el que el módulo de codificación aritmética 510 recibe valores de bin mediante la ruta normal, el módulo de modelado de contexto 506 puede proporcionar una variable de contexto (por ejemplo, un estado de contexto), de tal manera que el motor de codificación normal 508 puede realizar codificación aritmética basándose en las asignaciones de contexto proporcionadas por el módulo de modelado de contexto 506. En un ejemplo, el módulo de codificación aritmética 510 puede codificar una parte de prefijo de una cadena de bits usando una asignación de contexto y puede codificar una parte de sufijo de una cadena de bits sin usar asignaciones de contexto. Las asignaciones de contexto pueden definirse según los ejemplos descritos anteriormente con respecto a las tablas 2-13. Los modelos de contexto pueden almacenarse en memoria. El módulo de modelado de contexto 506 puede incluir una serie de tablas indexadas y/o utilizar funciones de mapeo para determinar un contexto y una variable de contexto para un bin particular. Después de codificar un valor de bin, el motor de codificación normal 508 puede actualizar un contexto basándose en los valores de bin reales y emitir el valor de bin codificado como parte de un flujo de bits. De esta manera, el módulo de codificación de entropía se configura para codificar uno o más elementos sintácticos basándose en las técnicas de asignación de contexto descritas en el presente documento.

35

40

La figura 6 es un diagrama de flujo que ilustra un método a modo de ejemplo para determinar un contexto para un valor de cadena binaria que indica la posición del último coeficiente significativo según las técnicas de esta divulgación. El método descrito en la figura 6 puede realizarse por cualquiera de los codificadores entrópicos o codificadores de vídeo a modo de ejemplo descritos en el presente documento. En la etapa 602, se obtiene una cadena binaria que indica la posición del último coeficiente de transformada significativo dentro de un bloque de vídeo. La cadena binaria puede definirse según el esquema de binarización descrito con respecto a la tabla 1. En la etapa 604, se determina un contexto para un valor de bin de la cadena binaria. Puede asignarse un contexto a un bin basándose en las técnicas descritas en el presente documento. El contexto puede determinarse por un codificador de vídeo o entrópico que accede a una tabla de consulta o realiza una función de mapeo. El contexto puede usarse para derivar una variable de contexto particular para un bin particular. Una variable de contexto puede ser un valor de binario de 7 bits que indica una de 64 probabilidades (estados) posibles y el estado más probable (por ejemplo, "1" o "0"). Tal como se describió anteriormente, en algunos casos, los bins pueden compartir contextos según las funciones de mapeo y las tablas 2-13 descritas anteriormente. En la etapa 606, se codifica un valor de bin usando un procedimiento de codificación aritmética que utiliza una variable de contexto, tal como CABAC. Obsérvese que cuando los bins comparten contextos, el valor de un bin puede afectar al valor de una variable de contexto usada para codificar un bin posterior según técnicas de codificación adaptativa según el contexto. Por ejemplo, si un bin particular es "1", un bin posterior puede codificarse basándose en una probabilidad aumentada de ser 1. De esta manera, la codificación de entropía de la cadena binaria puede incluir actualizar un estado de contexto de un modelo de contexto. Además, obsérvese que en algunos ejemplos puede inicializarse un modelo de

45

50

55

60

contexto para un nivel de segmento, de tal manera que los valores de los bins dentro de un segmento puede que no afecten a la codificación de bins dentro de un segmento posterior.

La figura 7 es un diagrama de bloques que ilustra un ejemplo de un decodificador de vídeo 30 que puede usar técnicas de codificación de coeficientes de transformada según se describe en esta divulgación. Por ejemplo, el decodificador de vídeo 30 representa un ejemplo de un decodificador de vídeo configurado para obtener una cadena binaria codificada que indica la posición del último coeficiente significativo dentro de un bloque de vídeo, en el que la cadena binaria codificada se codifica usando CABAC; determinar un contexto para un índice binario de la cadena binaria codificada basándose en el tamaño de bloque de vídeo, en el que el contexto se asigna a al menos dos índices binarios, en el que cada uno de los al menos dos índices binarios se asocian con tamaños de bloque de vídeo diferentes; y decodificar la cadena binaria codificada usando CABAC basada al menos en parte en el contexto determinado.

En el ejemplo de la figura 7, el decodificador de vídeo 30 incluye un módulo de decodificación de entropía 70, un módulo de compensación de movimiento 72, un módulo de predicción intra 74, un módulo de cuantificación inversa 76, un módulo de transformación inversa 78, una memoria intermedia de fotograma de referencia 82 y un sumador 80. El decodificador de vídeo 30 puede realizar, en algunos ejemplos, un paso de decodificación generalmente recíproco con respecto al paso de codificación descrito con respecto al codificador de vídeo 20.

El módulo de decodificación de entropía 70 realiza un procedimiento de decodificación de entropía con el flujo de bits codificado para recuperar una matriz unidimensional de coeficientes de transformada. El procedimiento de decodificación de entropía usado depende de la codificación de entropía usada por el codificador de vídeo 20 (por ejemplo, CABAC, CAVLC, etc.). El procedimiento de codificación de entropía usado por el codificador puede señalizarse en el flujo de bits codificado o puede ser un procedimiento predeterminado.

En algunos ejemplos, el módulo de decodificación de entropía 70 (o el módulo de cuantificación inversa 76) puede realizar una exploración de los valores recibidos usando una exploración que imita al modo de exploración usado por el módulo de codificación de entropía 56 (o el módulo de cuantificación 54) del codificador de vídeo 20. Aunque la exploración de coeficientes puede realizarse en el módulo de cuantificación inversa 76, la exploración se describirá con propósitos ilustrativos como que se realiza por el módulo de decodificación de entropía 70. Además, aunque se muestran como módulos funcionales independientes para facilidad de ilustración, la estructura y funcionalidad del módulo de decodificación de entropía 70, el módulo de cuantificación inversa 76, y otros módulos del decodificador de vídeo 30 pueden estar altamente integrados entre sí.

El módulo de cuantificación inversa 76 somete a cuantificación inversa, es decir, descuantifica, los coeficientes de transformada cuantificados proporcionados en el flujo de bits y decodificados por el módulo de decodificación de entropía 70. El procedimiento de cuantificación inversa puede incluir un procedimiento convencional, por ejemplo, similar a los procedimientos propuestos para HEVC o definido por la norma de decodificación H.264. El procedimiento de cuantificación inversa puede incluir el uso de un parámetro de cuantificación QP calculado por el codificador de vídeo 20 para la CU para determinar el grado de cuantificación y, asimismo, el grado de cuantificación inversa que debería aplicarse. El módulo de cuantificación inversa 76 puede someter a cuantificación inversa los coeficientes de transformada o bien antes o bien después de que se conviertan los coeficientes de una matriz unidimensional a una matriz bidimensional.

El módulo de transformada inversa 78 aplica una transformada inversa a los coeficientes de transformada inversa cuantificados. En algunos ejemplos, el módulo de transformada inversa 78 puede determinar una transformada inversa basándose en una señalización desde el codificador de vídeo 20, o deduciendo la transformada a partir de una o más características de codificación tales como tamaño de bloque, modo de codificación, o similar. En algunos ejemplos, el módulo de transformada inversa 78 puede determinar una transformada a aplicar al bloque actual basándose en una transformada señalizada en el nodo raíz de un árbol cuaternario para una LCU que incluye el bloque actual. Alternativamente, la transformada puede señalizarse en la raíz de un árbol cuaternario de TU para una CU de nodo hoja en el árbol cuaternario de CU. En algunos ejemplos, el módulo de transformada inversa 78 puede aplicar una transformada inversa en cascada, en la que el módulo de transformada inversa 78 aplica dos o más transformadas inversas a los coeficientes de transformada del bloque actual que está decodificándose.

El módulo de predicción intra 74 puede generar datos de predicción para un bloque actual de un fotograma actual basándose en un modo de predicción intra señalizado y datos procedentes de bloques decodificados previamente del fotograma actual. El módulo de compensación de movimiento 72 puede recuperar el vector de movimiento, la dirección de predicción de movimiento y el índice de referencia del flujo de bits codificado. La dirección de predicción de referencia indica si el modo de predicción inter es unidireccional (por ejemplo, un fotograma P) o bidireccional (un fotograma B). El índice de referencia indica en qué fotograma de referencia se basa el vector de movimiento candidato. Basándose en la dirección de predicción de movimiento, el índice de fotograma de referencia y el vector de movimiento recuperados, el módulo de compensación de movimiento 72 produce un bloque con movimiento compensado para la parte actual. Estos bloques con movimiento compensado recrean esencialmente el bloque predictivo usado para producir los datos residuales.

El módulo de compensación de movimiento 72 puede producir los bloques con movimiento compensado, realizando

posiblemente una interpolación basándose en filtros de interpolación. Los identificadores para los filtros de interpolación que van a usarse para la estimación de movimiento con precisión inferior a píxel pueden incluirse en los elementos sintácticos. El módulo de compensación de movimiento 72 puede usar filtros de interpolación tal como se usan por el codificador de vídeo 20 durante la codificación del bloque de vídeo para calcular valores interpolados para píxeles enteros secundarios de un bloque de referencia. El módulo de compensación de movimiento 72 puede determinar los filtros de interpolación usados por el codificador de vídeo 20 según la información sintáctica recibida y usar los filtros de interpolación para producir bloques predictivos.

Adicionalmente, el módulo de compensación de movimiento 72 y el módulo de predicción intra 74, en un ejemplo de HEVC, puede usar parte de la información sintáctica (por ejemplo, proporcionada por un árbol cuaternario) para determinar tamaños de LCU usados para codificar fotograma(s) de la secuencia de vídeo codificada. El módulo de compensación de movimiento 72 y el módulo de predicción intra 74 también pueden usar información sintáctica para determinar información de división que describe cómo se divide cada CU de un fotograma de la secuencia de vídeo codificada (y asimismo, cómo se dividen las CU secundarias). La información sintáctica también puede incluir modos que indican cómo se codifica cada división (por ejemplo, predicción intra o inter, y para la predicción intra un modo de codificación de predicción intra), uno o más fotogramas de referencia (y/o listas de referencia que contienen identificadores para los fotogramas de referencia) para cada PU sometida a codificación inter, y otra información para decodificar la secuencia vídeo codificada.

El sumador 80 combina los bloques residuales con los bloques de predicción correspondientes generados por el módulo de compensación de movimiento 72 o el módulo de predicción intra 74 para formar bloques decodificados. Si se desea, también puede aplicarse un filtro de desbloqueo para filtrar los bloques decodificados para eliminar artefactos con capacidad de bloqueo. Los bloques de vídeo decodificados se almacenan entonces en la memoria intermedia de fotograma de referencia 82, que proporciona bloques de referencia para una compensación de movimiento posterior y también produce vídeo decodificado para su presentación en un dispositivo de visualización (tal como el dispositivo de visualización 32 de la figura 1). La memoria intermedia de fotograma de referencia 82 también puede denominarse DPB.

La figura 8 es un diagrama de bloques que ilustra un módulo de decodificación de entropía 70 a modo de ejemplo que puede implementar las técnicas descritas en esta divulgación. El módulo de decodificación de entropía 70 recibe un flujo de bits sometido a codificación de entropía y decodifica elementos sintácticos a partir del flujo de bits. En un ejemplo, los elementos sintácticos pueden representar la posición del último coeficiente de transformada significativo dentro de un bloque de coeficientes de transformada. Los elementos sintácticos pueden incluir un elemento sintáctico que especifica una coordenada x de la posición del último coeficiente significativo dentro de un bloque de coeficiente de transformada y un elemento sintáctico que especifica una coordenada y de la posición del último coeficiente significativo dentro de un bloque de coeficiente de transformada. En un ejemplo, el módulo de decodificación de entropía 70 ilustrado en la figura 8 puede ser un decodificador CABAC. El módulo de decodificación de entropía 70 a modo de ejemplo en la figura 8 incluye un módulo de decodificación aritmética 702, que puede incluir un motor de decodificación de derivación 704 y un motor de decodificación normal 706. El módulo de decodificación de entropía 70 a modo de ejemplo también incluye una unidad de modelado de contexto 708 y un módulo de binarización inversa 710. El módulo de decodificación de entropía 70 a modo de ejemplo puede realizar las funciones recíprocas del módulo de codificación de entropía 56 a modo de ejemplo descrito con respecto a la figura 5. De esta manera, el módulo de decodificación de entropía 70 puede realizar una decodificación de entropía basándose en las técnicas de asignación de contexto descritas en el presente documento.

El módulo de decodificación aritmética 702 recibe un flujo de bits codificado. Tal como se muestra en la figura 8, el módulo de decodificación aritmética 702 puede procesar los valores de bin codificados según una ruta de derivación o la ruta de codificación normal. Puede señalizarse una indicación de si un valor de bin codificado debe procesarse según una ruta de derivación o puede señalizarse un paso normal en el flujo de bits con sintaxis de nivel superior. De manera consistente con el procedimiento CABAC descrito anteriormente, en el caso en el que el módulo de decodificación aritmética 702 recibe valores de bin procedentes de una ruta de derivación, el motor de decodificación de derivación 704 puede realizar decodificación aritmética con valores de bin sin utilizar un contexto asignado a un valor de bin. En un ejemplo, el motor de decodificación de derivación 704 puede suponer probabilidades iguales para valores posibles de un bin.

En el caso en el que módulo de decodificación aritmética 702 recibe valores de bin mediante la ruta normal, el módulo de modelado de contexto 708 puede proporcionar una variable de contexto, de tal manera que el motor de decodificación normal 706 pueda realizar decodificación aritmética basándose en las asignaciones de contexto proporcionadas por el módulo de modelado de contexto 708. Las asignaciones de contexto pueden definirse según los ejemplos descritos anteriormente con respecto a las tablas 2-13. Los modelos de contexto pueden almacenarse en memoria. El módulo de modelado de contexto 708 puede incluir una serie de tablas indexadas y/o utilizar funciones de mapeo para determinar un contexto y una parte de variable de contexto de un flujo de bits codificado. Después de decodificar un valor de bin, el motor de decodificación normal 706 puede actualizar un contexto basándose en los valores de bin decodificados. Además, el módulo de binarización inversa 710 puede realizar una binarización inversa con un valor de bin y usar una función de coincidencia de bin para determinar si un valor de bin es válido. El módulo de binarización inversa 710 también puede actualizar el módulo de modelado de contexto basándose en la determinación de coincidencia. Por tanto, el módulo de binarización inversa 710 emite elementos

sintácticos según una técnica de decodificación adaptativa según el contexto. De esta manera, el módulo de decodificación de entropía 70 se configura para decodificar uno o más elementos sintácticos basándose en las técnicas de asignación de contexto descritas en el presente documento.

5 La figura 9 es un diagrama de flujo que ilustra un método a modo de ejemplo para determinar un valor que indica la posición del último coeficiente significativo dentro de un coeficiente de transformada a partir de una cadena binaria según las técnicas de esta divulgación. El método descrito en la figura 9 puede realizarse por cualquiera de los decodificadores de vídeo o unidades de decodificación de entropía a modo de ejemplo descritos en el presente documento. En la etapa 902, se obtiene un flujo de bits codificado. Puede recuperarse un flujo de bits codificado desde una memoria o mediante una transmisión. El flujo de bits codificado puede codificarse según un procedimiento de codificación CABAC u otro procedimiento de codificación de entropía. En la etapa 904, se determina un contexto para una parte de la cadena binaria codificada. Puede asignarse un contexto a un bin codificado basándose en las técnicas descritas en el presente documento. El contexto puede determinarse por un decodificador de vídeo o entrópico que accede a una tabla de consulta o realiza una función de mapeo. El contexto puede determinarse basándose en la sintaxis de nivel superior proporcionada en el flujo de bits codificado. El contexto puede usarse para derivar una variable de contexto particular para un bin codificado particular. Tal como se describió anteriormente, una variable de contexto puede ser un valor binario de 7 bits que indica una de 64 probabilidades (estados) posibles y el estado más probable (por ejemplo, "1" o "0") y en algunos casos, los bins pueden compartir contextos. En la etapa 906, se decodifica una cadena binaria usando un procedimiento de decodificación aritmética que utiliza una variable de contexto, tal como CABAC. Puede decodificarse una cadena binaria bin a bin, en la que un modelo de contexto se actualiza después de decodificar cada bin. El flujo de bits decodificado puede incluir elementos sintácticos que se usan adicionalmente para decodificar coeficientes de transformada asociados con datos de vídeo codificados. De esta manera, la asignación de contextos a bins particulares utilizando las técnicas descritas anteriormente puede proporcionar una decodificación o codificación eficaz de datos de vídeo.

25 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 por, como una o más instrucciones o código, un medio legible por ordenador y ejecutarse por 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 de un sitio 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 legibles por ordenador tangibles que no son transitorios o (2) un medio de comunicación tal como una onda portadora o señal. Los medios de almacenamiento de datos pueden ser medios disponibles a los que pueden acceder 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.

40 A modo de ejemplo, y no de limitación, tales medios 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 estructuras de datos o instrucciones y al que pueda acceder un ordenador. Además, cualquier conexión se denomina con propiedad medio legible por ordenador. Por ejemplo, si se transmiten instrucciones desde un sitio web, servidor, u otra fuente remota usando un cable coaxial, cable de fibra óptica, par trenzado, línea de abonado digital (DSL), o tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, cable de fibra óptica, par trenzado, DSL, o tecnologías inalámbricas tales como infrarrojos, radio y microondas se incluyen en la definición de medio. Sin embargo, debe entenderse que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales, u otros medios transitorios, sino que en su lugar se refieren a medios de almacenamiento no transitorios, tangibles. Disco, tal como se usa en el presente documento, incluye disco compacto (CD), disco láser, disco óptico, disco versátil digital (DVD), disquete y disco de Blu-ray, en los que unos discos reproducen habitualmente datos de manera magnética, mientras que otros discos reproducen datos de manera óptica con láseres. También deben incluirse combinaciones de los anteriores dentro del alcance de los medios legibles por ordenador.

55 Pueden ejecutarse instrucciones por uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de uso general, circuitos integrados para aplicaciones específicas (ASIC), matrices de puertas programables *in situ* (FPGA), u otro conjunto de circuitos lógico integrado o discreto equivalente. Por consiguiente, el término "procesador", tal como se usa en el presente documento, puede referirse a cualquiera de las estructuras anteriores o cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento puede proporcionarse dentro de módulos de hardware y/o software dedicados configurados para la codificación y decodificación, o incorporados en un códec combinado. Además, las técnicas podrían implementarse completamente en uno o más circuitos o elementos lógicos.

Las técnicas de esta divulgación pueden implementarse en una amplia variedad de dispositivos o aparatos,

- 5 incluyendo un terminal inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). Diversos componentes, módulos o unidades se describen en esta divulgación para enfatizar aspectos funcionales de los dispositivos configurados para realizar las técnicas dadas a conocer, pero no requieren necesariamente su realización por unidades de hardware diferentes. Más bien, tal como se describió anteriormente, pueden combinarse diversas unidades en una unidad de hardware de códec o proporcionarse mediante una colección de unidades de hardware interoperativas, incluyendo uno o más procesadores tal como se describió anteriormente, junto con software y/o firmware adecuados.

REIVINDICACIONES

1. Método de codificación de coeficientes de transformada para un bloque de transformada de un bloque de vídeo, comprendiendo el método:
- 5 obtener (602) una cadena binaria que indica la posición de un último coeficiente significativo dentro del bloque de transformada de coeficientes de transformada;
- determinar (604) un contexto para un índice binario de la cadena binaria basándose en el tamaño del bloque de transformada; y
- codificar (606) la cadena binaria usando codificación aritmética binaria adaptativa según el contexto, CABAC, basada al menos en parte en el contexto determinado;
- 10 caracterizado porque
- el contexto asignado al índice binario de la cadena binaria es el mismo que el contexto asignado a al menos un índice binario de una cadena binaria que indica la posición del último coeficiente significativo dentro de un bloque de transformada de un tamaño diferente.
2. Método según la reivindicación 1, en el que el contexto se asigna a los índices binarios según un índice de contexto, `cbc_index`, definido por la función:
- 15
$$\text{ctx_index} = (n \gg \log_2(T) - 2) + 15;$$
- donde `n` es el índice binario y `T` es una dimensión del bloque de transformada.
3. Método según la reivindicación 1 ó 2, en el que el índice binario de la cadena binaria es el último índice binario de la cadena binaria de un bloque de transformada de 16x16 y el índice binario de la cadena binaria que indica la posición del último coeficiente significativo dentro de un bloque de transformada de un tamaño diferente es el último índice binario de un bloque de transformada de 32x32.
- 20 4. Método según una de las reivindicaciones 1 a 3, en el que la codificación de la cadena binaria usando CABAC basada al menos en parte en el contexto determinado incluye actualizar el contexto basándose en el valor de la cadena binaria, y en el que el bloque de transformada es un primer bloque de transformada; y que comprende además:
- 25 obtener una segunda cadena binaria que indica la posición de un último coeficiente significativo dentro de un segundo bloque de transformada, en el que el primer bloque de transformada y el segundo bloque de transformada tienen tamaños diferentes; y
- codificar la segunda cadena binaria usando CABAC basada al menos en parte en el contexto actualizado.
- 30 5. Método de decodificación de coeficientes de transformada para un bloque de transformada de un bloque de vídeo, comprendiendo el método:
- obtener (902) una cadena binaria codificada que indica la posición de un último coeficiente significativo dentro del bloque de transformada de coeficientes de transformada, en el que la cadena binaria codificada se codifica usando codificación aritmética binaria adaptativa según el contexto, CABAC;
- 35 determinar (904) un contexto para un índice binario de la cadena binaria codificada basándose en el tamaño del bloque de transformada; y
- decodificar (906) la cadena binaria codificada usando CABAC basada al menos en parte en el contexto determinado;
- caracterizado porque
- 40 el contexto asignado al índice binario de la cadena binaria es el mismo que el contexto asignado a al menos un índice binario de una cadena binaria que indica la posición de un último coeficiente significativo dentro de un bloque de transformada de un tamaño diferente.
6. Método según la reivindicación 5, en el que
- el contexto se asigna a los índices binarios según un índice de contexto, `ctx_index`, definido por la función:
- 45
$$\text{ctx_index} = (n \gg \log_2(T) - 2) + 15;$$
- en el que `n` es el índice binario y `T` es una dimensión del bloque de transformada.

7. Método según la reivindicación 5 ó 6, en el que el índice binario de la cadena binaria es el último índice binario de la cadena binaria de un bloque de transformada de 16x16 y el índice binario de la cadena binaria que indica la posición de un último coeficiente significativo dentro de un bloque de transformada de un tamaño diferente es el último índice binario de un bloque de transformada de 32x32.
- 5 8. Método según una de las reivindicaciones 5 a 7, en el que la decodificación de la cadena binaria codificada usando CABAC basada al menos en parte en el contexto determinado incluye actualizar el contexto basándose en el valor de la cadena binaria codificada, y en el que el bloque de transformada es un primer bloque de transformada; y que comprende además:
- 10 obtener una segunda cadena binaria codificada que indica la posición de un último coeficiente significativo dentro de un segundo bloque de transformada, en el que el primer bloque de transformada y el segundo bloque de transformada tienen tamaños diferentes; y
- decodificar la segunda cadena binaria codificada usando CABAC basada al menos en parte en el contexto actualizado.
- 15 9. Aparato configurado para codificar coeficientes de transformada para un bloque de transformada de un bloque de vídeo, que comprende:
- medios para obtener una cadena binaria que indica la posición de un último coeficiente significativo dentro del bloque de transformada de coeficientes de transformada;
- medios para determinar un contexto para un índice binario de la cadena binaria basándose en el tamaño del bloque de transformada; y
- 20 medios para codificar (56) la cadena binaria usando codificación aritmética binaria adaptativa según el contexto, CABAC basada al menos en parte en el contexto determinado;
- caracterizado porque
- el contexto asignado al índice binario de la cadena binaria es el mismo que el contexto asignado a al menos un índice binario de una cadena binaria que indica la posición del último coeficiente significativo dentro de un bloque de transformada de un tamaño diferente.
- 25 10. Aparato según la reivindicación 9, en el que el contexto se asigna a los índices binarios según un índice de contexto (ctx_index) definido por la función:
- $$\text{ctx_index} = (n \gg \log_2(T) - 2) + 15;$$
- en el que n es el índice binario y T es una dimensión del bloque de transformada.
- 30 11. Aparato según la reivindicación 9 ó 10, en el que la codificación de la cadena binaria usando CABAC basada al menos en parte en el contexto determinado incluye actualizar el contexto basándose en el valor de la cadena binaria, y en el que el bloque de transformada es un primer bloque de transformada; y que comprende además:
- 35 medios para obtener una segunda cadena binaria que indica la posición del último coeficiente significativo dentro de un segundo bloque de transformada, en el que el primer bloque de transformada y el segundo bloque de transformada tienen tamaños diferentes; y
- medios para someter a codificación de entropía la segunda cadena binaria usando CABAC basada al menos en parte en el contexto actualizado.
- 40 12. Aparato configurado para decodificar coeficientes de transformada para un bloque de transformada de un bloque de vídeo, que comprende:
- medios para obtener una cadena binaria codificada que indica la posición de un último coeficiente significativo dentro del bloque de transformada de coeficientes de transformada, en el que la cadena binaria codificada se codifica usando codificación aritmética binaria adaptativa según el contexto, CABAC;
- 45 medios para determinar un contexto para un índice binario de la cadena binaria codificada basándose en el tamaño del bloque de transformada; y
- medios para decodificar (70) la cadena binaria codificada usando CABAC basada al menos en parte en el contexto determinado;
- caracterizado porque

el contexto asignado al índice binario de la cadena binaria es el mismo que el contexto asignado a al menos un índice binario de una cadena binaria que indica la posición de un último coeficiente significativo dentro de un bloque de transformada de un tamaño diferente.

- 5 13. Aparato según la reivindicación 12, en el que el contexto se asigna a los índices binarios según un índice de contexto, `ctx_index`, definido por la función:

$$\text{ctx_index} = (n \gg \log_2(T) - 2) + 15;$$

donde `n` es el índice binario y `T` es una dimensión del bloque de transformada.

- 10 14. Aparato según la reivindicación 12 ó 13, en el que la decodificación de la cadena binaria codificada usando CABAC basada al menos en parte en el contexto determinado incluye actualizar el contexto basándose en el valor de la cadena binaria codificada, y en el que el bloque de transformada es un primer bloque de transformada; y que comprende además:

medios para obtener una segunda cadena binaria codificada que indica la posición de un último coeficiente significativo dentro de un segundo bloque de transformada, en el que el primer bloque de transformada y el segundo bloque de transformada tienen tamaños diferentes; y

- 15 15. medios para decodificar la segunda cadena binaria codificada usando CABAC basada al menos en parte en el contexto actualizado.

- 20 15. Medio de almacenamiento legible por ordenador no transitorio que tiene instrucciones almacenadas en el mismo que, tras su ejecución, hacen que uno o más procesadores de un dispositivo de codificación de vídeo:

obtengan una cadena binaria que indica la posición del último coeficiente significativo dentro de un bloque de transformada de coeficientes de transformada de un bloque de vídeo;

determinen un contexto para un índice binario de la cadena binaria basándose en el tamaño del bloque de transformada; y

- 25 25. codifiquen la cadena binaria usando codificación aritmética binaria adaptativa según el contexto, CABAC, basada al menos en parte en el contexto determinado;

caracterizado porque

el contexto asignado al índice binario de la cadena binaria es el mismo que el contexto asignado a al menos un índice binario de una cadena binaria que indica la posición del último coeficiente significativo dentro de un bloque de transformada de un tamaño diferente.

30

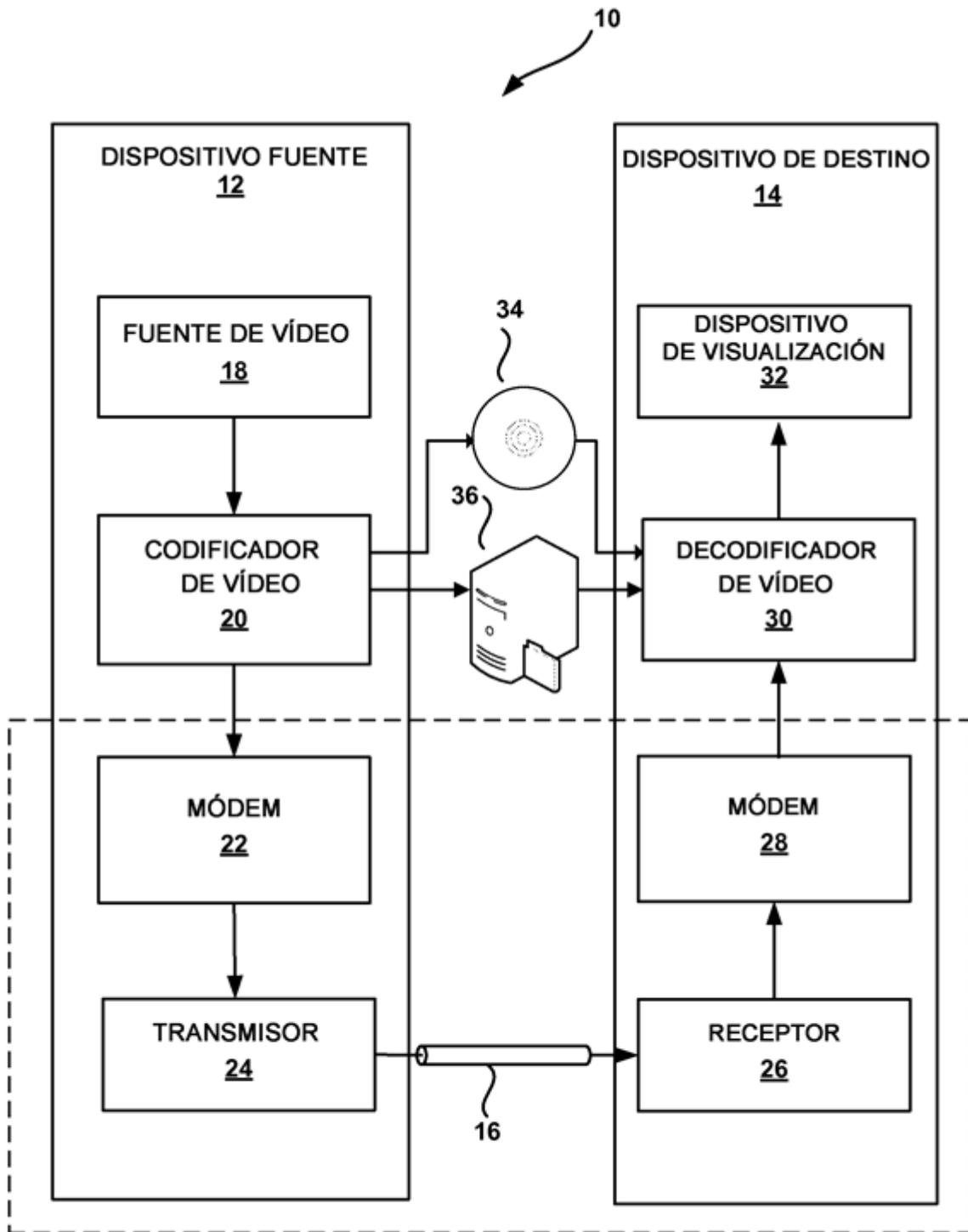


FIG. 1

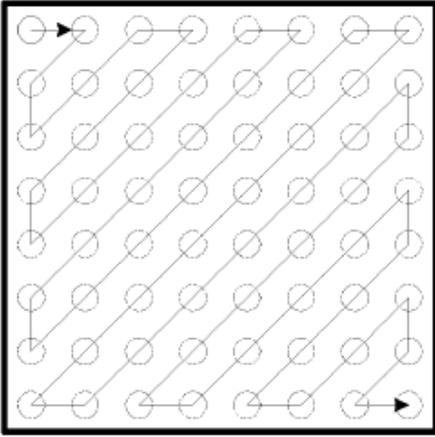


FIG. 2A

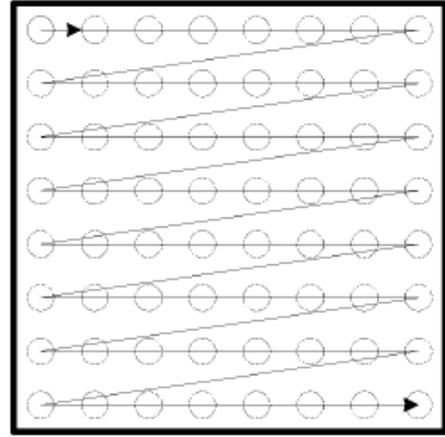


FIG. 2B

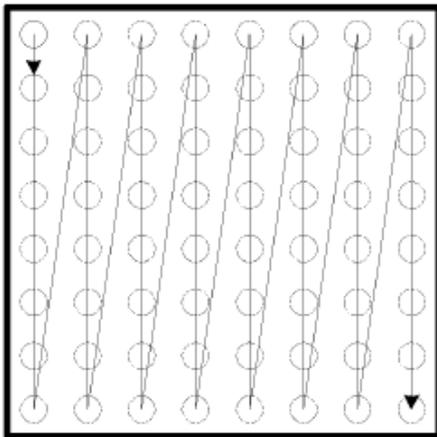


FIG. 2C

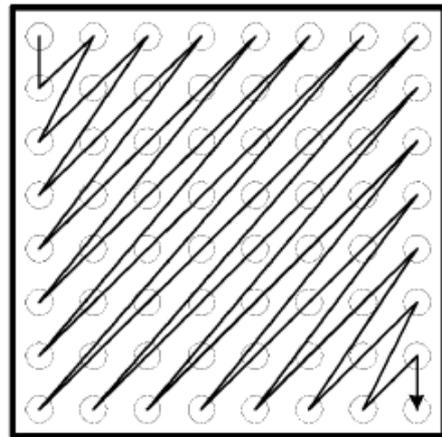


FIG. 2D

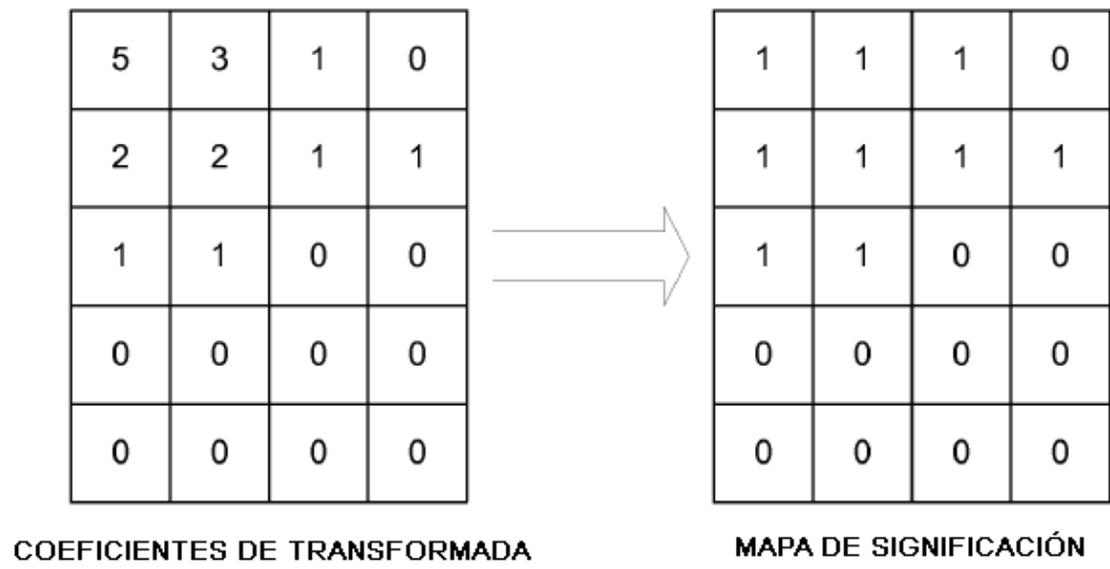


FIG. 3

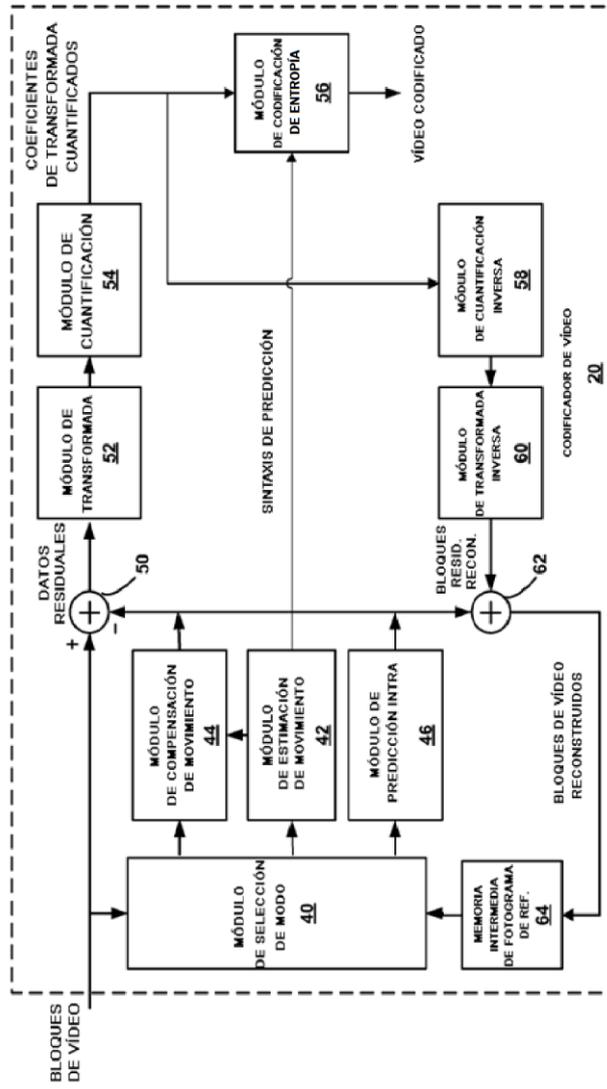


FIG. 4

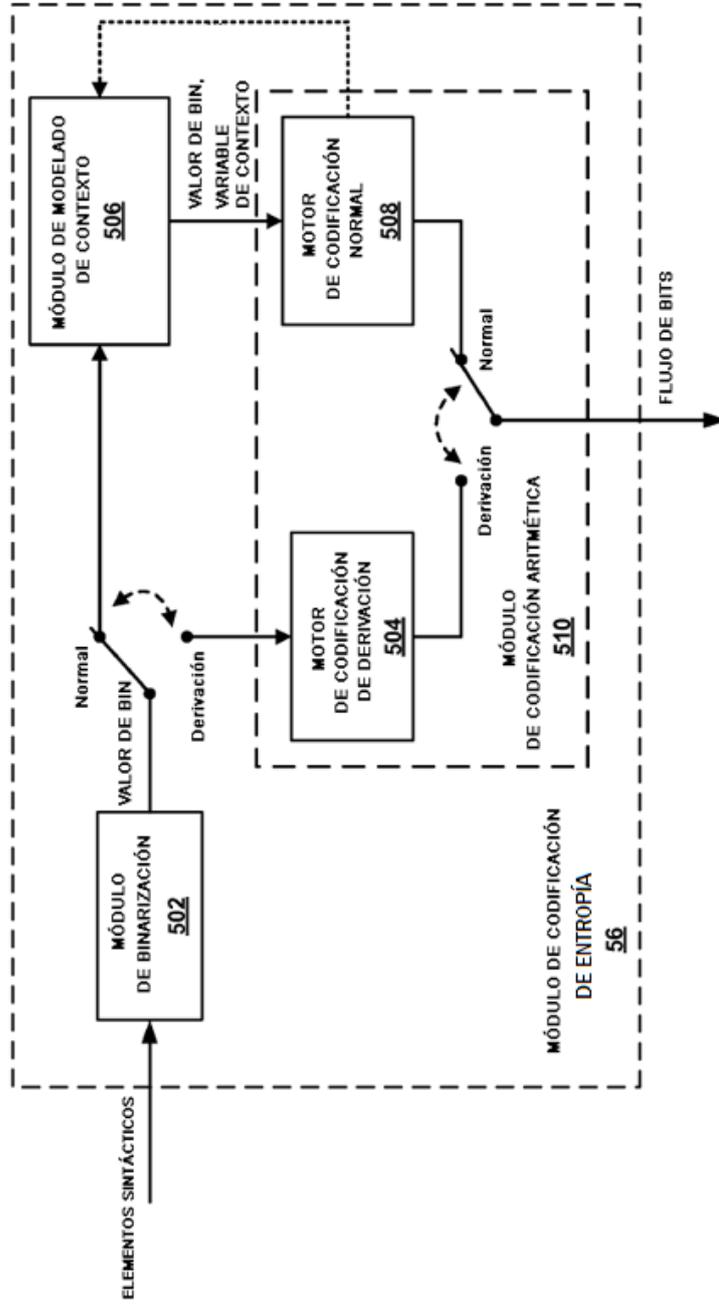


FIG. 5

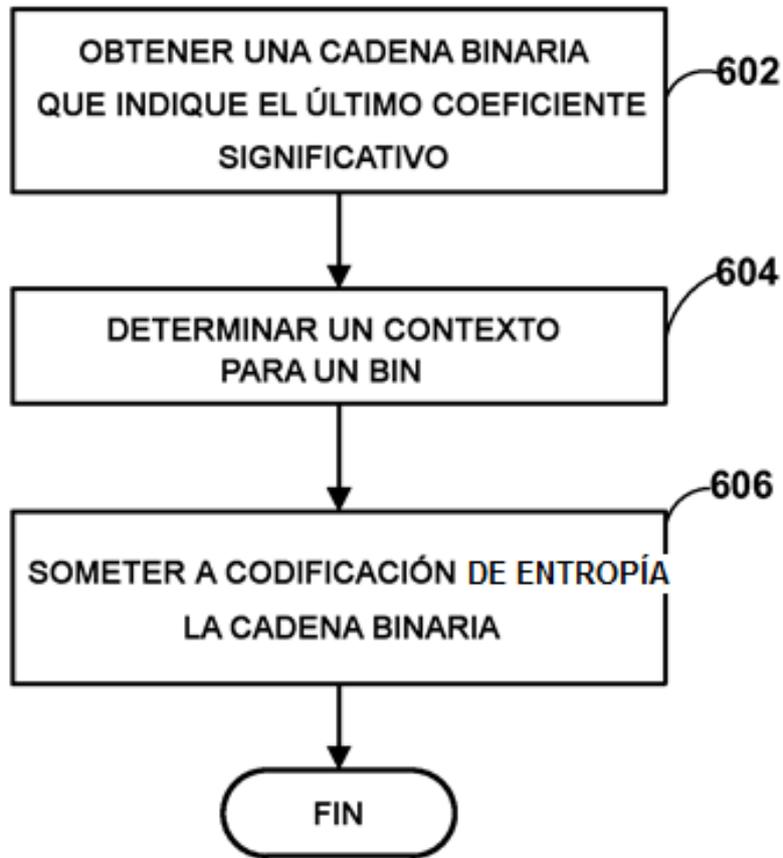


FIG. 6

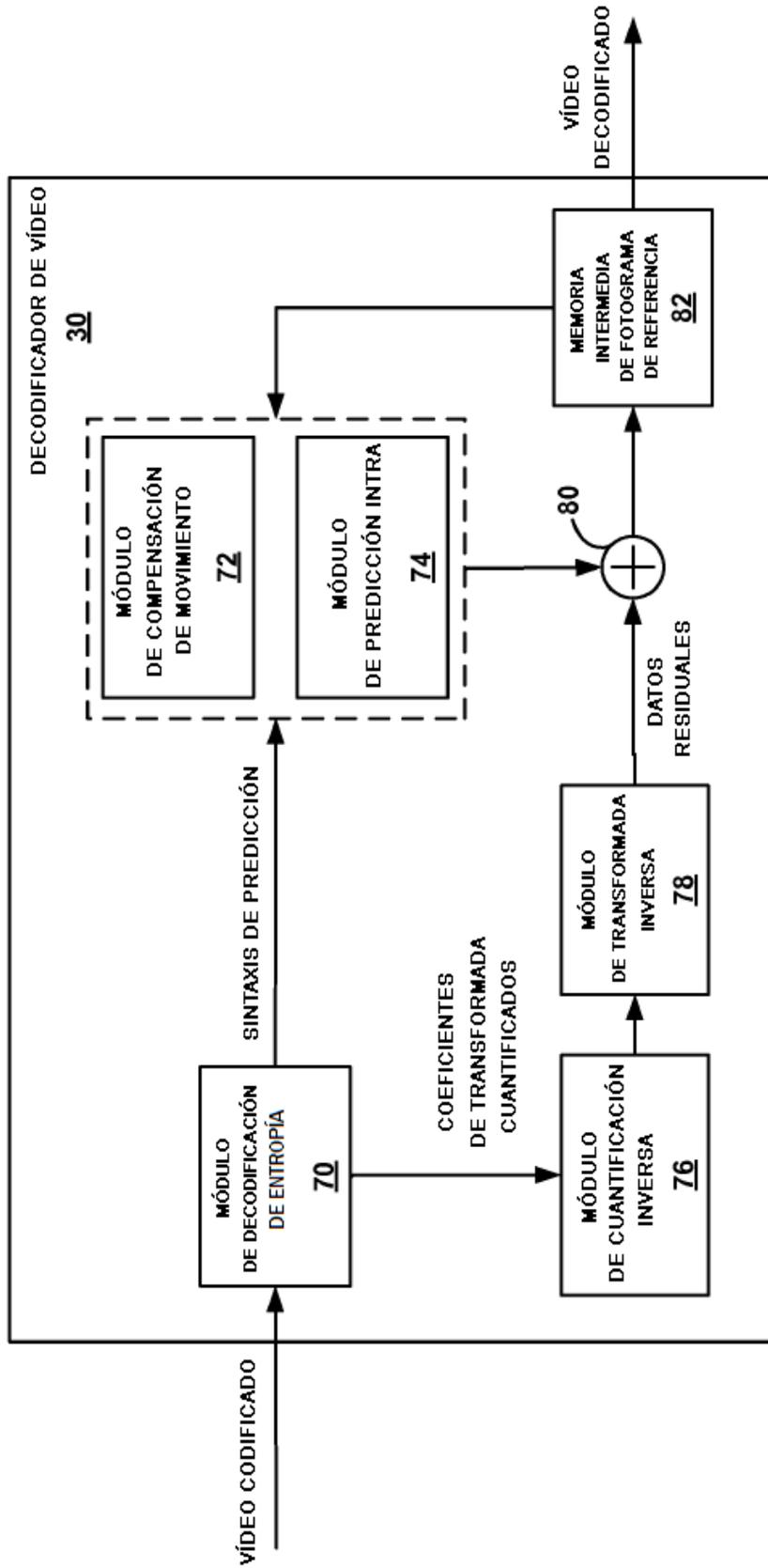


FIG. 7

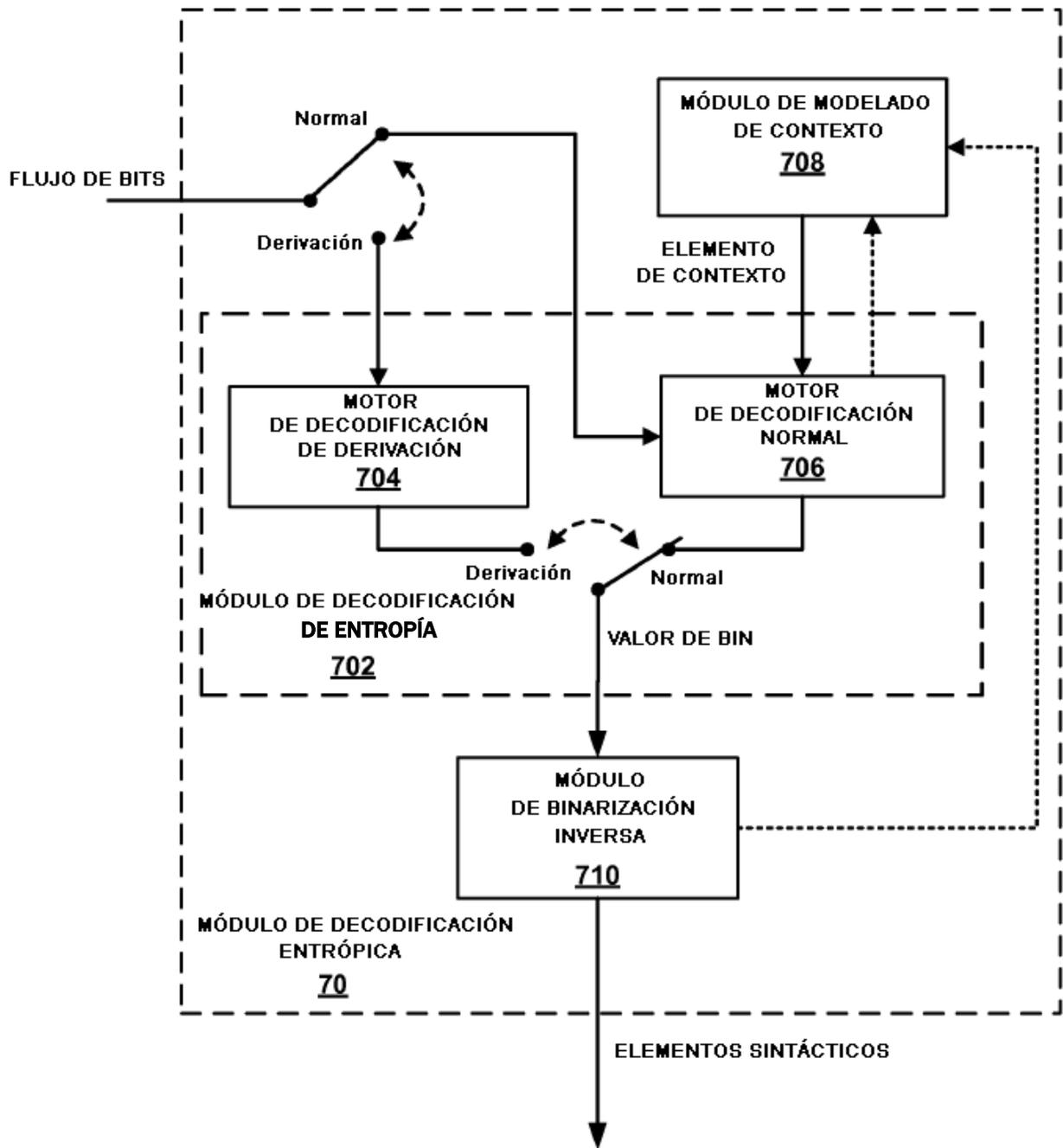


FIG. 8

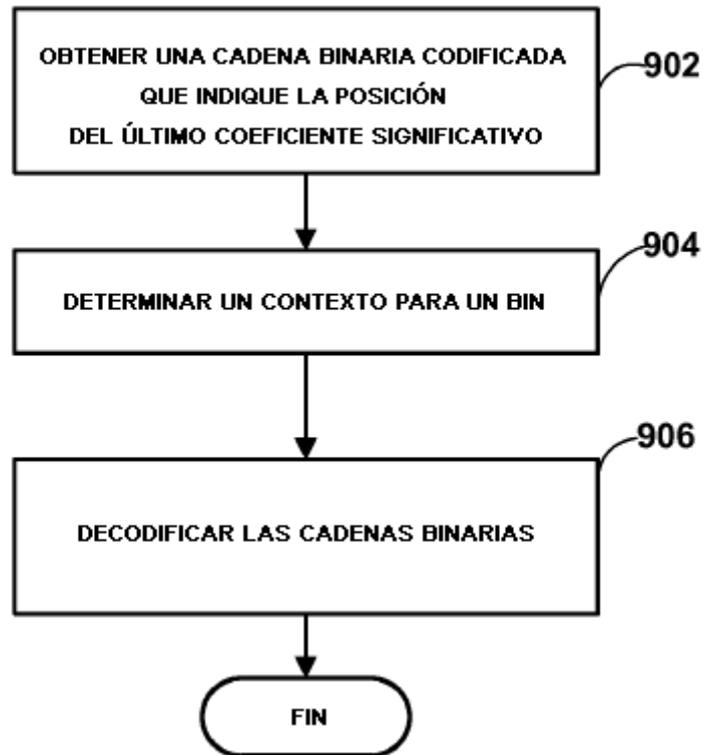


FIG. 9