

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 717 679**

51 Int. Cl.:

H04N 19/70 (2014.01)

H04N 19/13 (2014.01)

H04N 19/463 (2014.01)

H04N 19/18 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **06.11.2012 PCT/US2012/063707**

87 Fecha y número de publicación internacional: **16.05.2013 WO13070604**

96 Fecha de presentación y número de la solicitud europea: **06.11.2012 E 12788041 (7)**

97 Fecha y número de publicación de la concesión europea: **09.01.2019 EP 2777261**

54 Título: **Codificación progresiva de una posición de último coeficiente significativo**

30 Prioridad:

08.11.2011 US 201161557317 P

20.11.2011 US 201161561909 P

05.11.2012 US 201213669032

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

24.06.2019

73 Titular/es:

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

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

72 Inventor/es:

**CHIEN, WEI-JUNG;
SOLE ROJALS, JOEL;
KARCZEWICZ, MARTA y
JOSHI, RAJAN LAXMAN**

74 Agente/Representante:

MILTENYI , Peter

ES 2 717 679 T3

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

DESCRIPCIÓN

Codificación progresiva de una posición de último coeficiente significativo

5

CAMPO TÉCNICO

Esta divulgación se refiere a la codificación de vídeo.

ANTECEDENTES

- 10 Vadim Segeren ET AL: "Binarization modification for last position coding" JOINT COLLABORATIVE TEAM ON VIDEO CODING OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG. 16, 14 de julio de 2011, páginas 1-10, describe un procedimiento de codificación de vídeo HM3.1 que especifica una modificación de la codificación de la posición del último coeficiente significativo dentro de un bloque de vídeo y determina una primera cadena de bits para el valor que indica la posición del último coeficiente significativo en base a un esquema de codificación unaria truncada. El valor de la posición del último coeficiente significativo es representado en dos partes, en el que la primera parte del valor del componente es representada con un código unario truncado.

Además, este documento describe que el valor de la posición del último coeficiente significativo es un valor binario representado en dos partes, por lo que se aplica una representación binaria fija para la segunda parte.

20

Las capacidades de vídeo digital pueden incorporarse a una amplia gama de dispositivos, incluidos televisores digitales, sistemas de transmisión directa digital, sistemas de transmisión inalámbrica, asistentes digitales personales (PDA), sistemas informáticos portátiles o de escritorio, tabletas, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, radiotelefonos celulares o satelitales, denominados "teléfonos inteligentes", dispositivos de teleconferencia de vídeo, dispositivos de streaming de vídeo y otros. Los dispositivos de vídeo digital implementan técnicas de compresión de vídeo, tales como las descritas en los estándares definidos por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificación de vídeo avanzada (AVC), el estándar de codificación de vídeo de alta eficiencia (HEVC) actualmente en desarrollo, y las extensiones de dichos estándares. Los dispositivos de vídeo pueden transmitir, recibir, codificar, decodificar y/o almacenar información de vídeo digital de manera más eficiente implementando dichas técnicas de compresión de vídeo.

25

30

Las técnicas de compresión de vídeo realizan una predicción espacial (intra-imagen) y/o temporal (inter-imágenes) para reducir o eliminar la redundancia inherente en las secuencias de vídeo. Para la codificación de vídeo basada en bloques, un segmento de vídeo (es decir, una trama de vídeo o una parte de una trama de vídeo) se puede dividir en bloques de vídeo, que también se pueden denominar bloques de árbol, unidades de codificación (CU) y/o nodos de codificación. Los bloques de vídeo en un segmento (slice) intra-codificado (I) de una imagen se codifican utilizando la predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen. Los bloques de vídeo en un segmento inter-codificado (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 tramas y las imágenes de referencia pueden denominarse tramas de referencia.

35

40

La predicción espacial o temporal produce como resultado un bloque predictivo para un bloque a codificar. Los datos residuales representan diferencias de píxeles entre el bloque original a codificar y el bloque predictivo. Un bloque inter-codificado se codifica de acuerdo con un vector de movimiento que apunta a un bloque de muestras de referencia que forman el bloque predictivo, y los datos residuales que indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intra-codificado se codifica de acuerdo con un modo de intra-codificación y los datos residuales. Para una compresión adicional, los datos residuales pueden transformarse del dominio de píxeles a un dominio de transformación, produciendo como resultado coeficientes de transformación residuales, que luego pueden cuantificarse. Los coeficientes de transformación cuantificados, inicialmente dispuestos en una matriz bidimensional, pueden ser explorados con el fin de producir un vector unidimensional de coeficientes de transformación, y se puede aplicar la codificación de entropía para conseguir aún más compresión.

45

50

55 BREVE DESCRIPCIÓN DE LOS DIBUJOS

La figura 1 es un diagrama de bloques que ilustra un ejemplo de sistema de codificación y decodificación de vídeo que puede utilizar las técnicas descritas en esta divulgación.

- 60 Las figuras 2A - 2D ilustran ejemplos de órdenes de exploración de valores de coeficiente.

La figura 3 ilustra un ejemplo de un mapa de significancia relativo a un bloque de valores de coeficiente.

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

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

La figura 6 es un diagrama de flujo que ilustra un procedimiento de ejemplo para determinar una cadena de bits para un valor que indica la posición de un último coeficiente significativo de acuerdo con las técnicas de esta divulgación.

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

La figura 8 es un diagrama de flujo que ilustra un procedimiento de ejemplo para determinar un valor que indica la posición de un último coeficiente significativo a partir de una cadena de bits de acuerdo con las técnicas de esta divulgación.

RESUMEN

La presente invención se define en las reivindicaciones independientes. En general, esta divulgación describe técnicas para codificar datos de video. La codificación de video generalmente implica pronosticar un bloque de datos de video usando un modo de predicción particular, y codificar valores residuales para el bloque en función de diferencias entre el bloque pronosticado y el bloque real que se está codificando. Un bloque residual incluye dichas diferencias píxel por píxel. El bloque residual puede ser transformado y cuantificado. Un codificador de video puede incluir una unidad de cuantificación que mapea coeficientes de transformación con valores de nivel discretos. Esta divulgación proporciona técnicas para codificar la posición de un último coeficiente significativo dentro de un bloque de video.

25 En un ejemplo, un procedimiento para codificar datos de video comprende obtener un valor que indica una posición de un último coeficiente significativo dentro de un bloque de video de tamaño T, determinar una primera cadena de bits para el valor que indica la posición del último coeficiente significativo en base a un esquema de codificación unaria truncada definido por una longitud máxima de bits definida por $2\log_2(T)-1$, determinar una segunda cadena de bits para el valor que indica la posición del último coeficiente significativo en base a un esquema de codificación de longitud fija y codificar la primera y la segunda cadenas de bits en un flujo de bits.

30 En otro ejemplo, un procedimiento para decodificar datos de video comprende obtener de un flujo de bits codificado una primera cadena de bits y una segunda cadena de bits, determinar un valor que indica la posición de un último coeficiente significativo dentro de un bloque de video de tamaño T basándose en parte en la primera cadena de bits, en el que la primera cadena de bits es definida por un esquema de codificación unaria truncada con una longitud máxima de bits definida por $2\log_2(T)-1$, y determinar el valor que indica la posición del último coeficiente significativo basándose en parte en la segunda cadena de bits, en el que la segunda cadena de bits es definida por un esquema de codificación de longitud fija.

40 En otro ejemplo, un aparato para codificar datos de video comprende un dispositivo de codificación de video configurado para obtener un valor que indica una posición de un último coeficiente significativo dentro de un bloque de video de tamaño T, determinar una primera cadena de bits para el valor que indica la posición del último coeficiente significativo en base a un esquema de codificación unaria truncada definido por una longitud máxima de bits definida por $2\log_2(T)-1$, determinar una segunda cadena de bits para el valor que indica la posición del último coeficiente significativo en base a un esquema de codificación de longitud fija, y codificar la primera y segunda cadenas de bits en un flujo de bits.

50 En otro ejemplo, un aparato para decodificar datos de video comprende un dispositivo de decodificación de video configurado para obtener de un flujo de bits (bitstream) codificado una primera cadena de bits y una segunda cadena de bits, determinar un valor que indica la posición de un último coeficiente significativo dentro de un bloque de video de tamaño T basándose en parte en la primera cadena de bits, en el que la primera cadena de bits es definida por un esquema de codificación unaria truncada con una longitud máxima de bits definida por $2\log_2(T)-1$, y determinar el valor que indica la posición del último coeficiente significativo basándose en parte en la segunda cadena de bits, en el que la segunda cadena de bits es definida por un esquema de codificación de longitud fija.

60 En otro ejemplo, un dispositivo para codificar datos de video comprende medios para obtener un valor que indica una posición de un último coeficiente significativo dentro de un bloque de video de tamaño T, medios para determinar una primera cadena de bits para el valor que indica la posición del último coeficiente significativo en base a un esquema de codificación unaria truncada definido por una longitud máxima de bits definida por $2\log_2(T)-1$, medios para determinar una segunda cadena de bits para el valor que indica la posición del último coeficiente significativo en base a un esquema de codificación de longitud fija, y medios para codificar la primera y segunda cadenas de bits en un flujo de bits.

En otro ejemplo, un dispositivo para decodificar datos de video comprende medios para obtener de un flujo de bits codificado una primera cadena de bits y una segunda cadena de bits, medios para determinar un valor que indica la posición de un último coeficiente significativo dentro de un bloque de video de tamaño T basándose en parte en la primera cadena de bits, en el que la primera cadena de bits es definida por un esquema de codificación unaria truncada con una longitud máxima de bits definida por $2\log_2(T)-1$, y medios para determinar el valor que indica la posición del último coeficiente significativo basándose en parte en la segunda cadena de bits, en el que la segunda cadena de bits es definida por un esquema de codificación de longitud fija.

10 En otro ejemplo, un medio de almacenamiento legible informáticamente comprende instrucciones almacenadas en el mismo que, cuando son ejecutadas, hacen que un procesador de un dispositivo para codificar datos de video haga que uno o más procesadores obtengan un valor que indica una posición de un último coeficiente significativo dentro de un bloque de video de tamaño T, determinen una primera cadena de bits para el valor que indica la posición del último coeficiente significativo en base a un esquema de codificación unaria truncada definido por una longitud máxima de bits definida por $2\log_2(T)-1$, determinen una segunda cadena de bits para el valor que indica la posición del último coeficiente significativo en base a un esquema de codificación de longitud fija, y codifiquen la primera y segunda cadenas de bits en un flujo de bits.

20 En otro ejemplo, un medio de almacenamiento legible informáticamente comprende instrucciones almacenadas en el mismo que, cuando son ejecutadas, hacen que un procesador de un dispositivo para decodificar datos de video obtenga de un flujo de bits codificado una primera cadena de bits y una segunda cadena de bits, determine un valor que indica la posición de un último coeficiente significativo dentro de un bloque de video de tamaño T basándose en parte en la primera cadena de bits, en el que la primera cadena de bits es definida por un esquema de codificación unaria truncada con una longitud máxima de bits definida por $2\log_2(T)-1$, y determine el valor que indica la posición del último coeficiente significativo basándose en parte en la segunda cadena de bits, en el que la segunda cadena de bits es definida por un esquema de codificación de longitud fija.

30 En un ejemplo, un procedimiento para decodificar datos de video comprende obtener de un flujo de bits codificado una primera cadena de bits y una segunda cadena de bits, determinar un valor que indica la posición de un último coeficiente significativo dentro de un bloque de video de tamaño T basándose en parte en la primera cadena de bits, en el que la primera cadena de bits es definida por un esquema de codificación unaria truncada con una longitud máxima de bits definida por $\log_2(T)+1$, y determinar el valor que indica la posición del último coeficiente significativo basándose en parte en la segunda cadena de bits, en el que la segunda cadena de bits es definida por un esquema de codificación de longitud fija.

35 En un ejemplo, un procedimiento para decodificar datos de video comprende obtener de un flujo de bits codificado una primera cadena de bits y una segunda cadena de bits, determinar un valor que indica la posición de un último coeficiente significativo dentro de un bloque de video de tamaño T basándose en parte en la primera cadena de bits, en el que la primera cadena de bits es definida por un esquema de codificación unaria truncada con una longitud máxima de bits definida por $\log_2(T)$, y determinar el valor que indica la posición del último coeficiente significativo basándose en parte en la segunda cadena de bits, en el que la segunda cadena de bits es definida por un esquema de codificación de longitud fija.

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

DESCRIPCIÓN DETALLADA

Esta divulgación proporciona técnicas para reducir la longitud de una cadena de bits utilizada para indicar la posición del último coeficiente significativo dentro de un bloque de coeficientes de transformación. La cadena de bits puede ser particularmente útil para la codificación aritmética binaria adaptativa al contexto (CABAC). En un ejemplo, se puede usar una estructura de palabra de código progresiva con un número reducido de valores binarios (bins) y códigos unarios truncados más cortos para indicar la posición del último coeficiente significativo. Además, en un ejemplo, reduciendo la longitud máxima del código unario truncado, también se puede reducir el número de modelos de contexto CABAC para la posición de último coeficiente significativo.

55 Un codificador de video puede estar configurado para determinar una primera y segunda cadenas de bits para un valor que indica la posición del último coeficiente significativo, dentro de un bloque de video de tamaño T. Un decodificador de video puede estar configurado para determinar un valor que indica la posición de un último coeficiente significativo dentro de un bloque de video de tamaño T en base a una primera y segunda cadenas de bits. En un ejemplo, la primera cadena de bits se puede basar en un esquema de codificación unaria truncada definido por una longitud máxima de bits definida por $2\log_2(T)-1$ y la segunda cadena de bits se puede basar en un esquema de codificación de longitud fija definido por una longitud máxima de bits definida por $\log_2(T)-2$. En otro ejemplo, la primera cadena de bits se puede basar en un esquema de codificación unaria truncada definido por una longitud máxima de bits definida por

$\log_2(T)+1$ y la segunda cadena de bits se puede basar en un esquema de codificación de longitud fija definido por una longitud máxima de bits definida por $\log_2(T)-1$. En aún otro ejemplo, la primera cadena de bits se puede basar en un esquema de codificación unaria truncada definido por una longitud máxima de bits definida por $\log_2(T)$ y la segunda cadena de bits se puede basar en un esquema de codificación de longitud fija definido por una longitud máxima de 5 bits definida por $\log_2(T)-1$.

La figura 1 es un diagrama de bloques que ilustra un ejemplo de sistema de codificación y decodificación de video 10 que puede utilizar las técnicas descritas en esta divulgación. Según se muestra en la figura 1, el sistema 10 incluye un dispositivo de origen 12 que genera datos de video codificados para su posterior decodificación por un dispositivo 10 de destino 14. El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender cualquiera de entre una amplia gama de dispositivos, incluidos sistemas informáticos de escritorio, sistemas informáticos notebook (es decir, portátiles), tabletas, decodificadores, teléfonos móviles tales como los denominados teléfonos "inteligentes", los denominados paneles (pads) "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de video juegos, dispositivo de streaming de video, o similares. En algunos casos, el 15 dispositivo de origen 12 y el dispositivo de destino 14 pueden estar equipados para comunicación inalámbrica.

El dispositivo de destino 14 puede recibir los datos de video codificados para su decodificación a través de un enlace 16. El enlace 16 puede comprender cualquier tipo de medio o dispositivo capaz de mover los datos de video codificados desde el dispositivo de origen 12 al dispositivo de destino 14. En un ejemplo, el enlace 16 puede comprender un medio 20 de comunicación para permitir que el dispositivo de origen 12 transmita datos de video codificados directamente al dispositivo de destino 14 en tiempo real. Los datos de video codificados pueden ser modulados de acuerdo con un estándar de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitidos al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrico o por cable, tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión físicas. El medio de comunicación puede 25 formar parte de una red basada en paquetes, tal como una red de área local, una red de área amplia o una red global tal como Internet. El medio de comunicación puede incluir enrutadores, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo de origen 12 al dispositivo de destino 14.

Alternativamente, una interfaz de salida 22 puede proporcionar datos codificados a un dispositivo de almacenamiento 32. De manera similar, una interfaz de entrada 28 puede acceder a datos codificados del dispositivo de almacenamiento 32. El dispositivo de almacenamiento 32 puede incluir cualquiera de entre una variedad de medios de almacenamiento de datos distribuidos o accedidos localmente tales como un disco duro, discos Blu-ray, DVD, CD-ROM, memoria flash, memoria volátil o no volátil, o cualquier otro medio de almacenamiento digital adecuado para 35 almacenar datos de video codificados. En un ejemplo adicional, el dispositivo de almacenamiento 32 puede corresponder a un servidor de archivos u otro dispositivo de almacenamiento intermedio que pueda contener el video codificado generado por el dispositivo de origen 12. El dispositivo de destino 14 puede acceder a los datos de video almacenados del dispositivo de almacenamiento 32 a través de transmisión o descarga. El servidor de archivos puede ser cualquier tipo de servidor capaz de almacenar datos de video codificados y transmitir esos datos de video 40 codificados al dispositivo de destino 14. Ejemplos de servidores de archivos incluyen un servidor web (por ejemplo, para un sitio web), un servidor FTP, dispositivos de almacenamiento conectados a la red (Network Attached Storage, NAS), o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de video codificados a través de cualquier conexión de datos estándar, incluida una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión WiFi), una conexión por cable (por ejemplo, DSL, cable módem, etc.), o una combinación 45 de ambos que sea adecuada para acceder a datos de video codificados almacenados en un servidor de archivos. La transmisión de datos de video codificados desde el dispositivo de almacenamiento 32 puede ser una transmisión streaming, una transmisión de descarga o una combinación de ambas.

Las técnicas de la presente divulgación no se limitan necesariamente a aplicaciones o configuraciones inalámbricas. 50 Las técnicas se pueden aplicar a la codificación de video en soporte a cualquiera de entre una variedad de aplicaciones multimedia, tales como transmisiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de video de streaming, por ejemplo, a través de Internet, codificación de video digital para almacenamiento en un medio de almacenamiento de datos, decodificación de video digital almacenado en un medio de almacenamiento de datos u otras aplicaciones. En algunos ejemplos, el sistema 10 puede estar 55 configurado para admitir la transmisión de video de una o dos vías para admitir aplicaciones tales como streaming de video, reproducción de video, difusión de video y/o telefonía de video.

En el ejemplo de la figura 1, el dispositivo de origen 12 incluye una fuente de video 18, un codificador de video 20 y una interfaz de salida 22. En algunos casos, la interfaz de salida 22 puede incluir un modulador/demodulador (módem) 60 y/o un transmisor. En el dispositivo de origen 12, la fuente de video 18 puede incluir una fuente tal como un dispositivo de captura de video, por ejemplo, una cámara de video, un archivo de video que contenga video capturado previamente, una interfaz de alimentación de video para recibir video de un proveedor de contenidos de video y/o una sistema de gráficos informáticos para generar datos de gráficos informáticos como fuente de video, o una combinación

de dichas fuentes. Como ejemplo, si la fuente de video 18 es una cámara de video, el dispositivo de origen 12 y el dispositivo de destino 14 pueden formar unos denominados teléfonos con cámara o video teléfonos. Sin embargo, las técnicas descritas en esta divulgación pueden ser aplicables a la codificación de video en general, y se pueden aplicar a aplicaciones inalámbricas y/o por cable.

5 El video capturado, pre-capturado, o generado informáticamente puede ser codificado por el codificador de vídeo 12. Los datos de vídeo codificados pueden ser transmitidos directamente al dispositivo de destino 14 a través de la interfaz de salida 22 del dispositivo de origen 20. Los datos de vídeo codificados pueden también (o alternativamente) almacenarse en el dispositivo de almacenamiento 32 para su posterior acceso por parte del dispositivo de destino 14
10 u otros dispositivos, para su decodificación y/o reproducción.

El dispositivo de destino 14 incluye una interfaz de entrada 28, un decodificador de video 30 y un dispositivo de visualización 32. En algunos casos, la interfaz de entrada 28 puede incluir un receptor y/o un módem. La interfaz de entrada 28 del dispositivo de destino 14 recibe los datos de video codificados a través del enlace 16. Los datos de
15 video codificados comunicados a través del enlace 16, o proporcionados en el dispositivo de almacenamiento 32, pueden incluir una variedad de elementos de sintaxis generados por el codificador de video 20 para su uso por un decodificador de video, tal como el decodificador de video 30, en la decodificación de los datos de video. Dichos elementos de sintaxis pueden incluirse con los datos de video codificados transmitidos en un medio de comunicación, almacenados en un medio de almacenamiento o almacenados en un servidor de archivos.

20 El dispositivo de visualización 32 puede estar integrado o ser externo al dispositivo de destino 14. En algunos ejemplos, el dispositivo de destino 14 puede incluir un dispositivo de visualización integrado y también puede estar configurado para interactuar 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 muestra los datos de video decodificados
25 a un usuario, y puede comprender cualquiera de entre 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.

El codificador de video 20 y el decodificador de video 30 pueden operar de acuerdo con un estándar de compresión
30 de video, tal como el estándar de codificación de video de alta eficiencia (HEVC) actualmente en desarrollo, y pueden ajustarse al modelo de prueba del HEVC (HM). Alternativamente, el codificador de video 20 y el decodificador de video 30 pueden operar de acuerdo con otros estándares propietarios o industriales, tales como el estándar ITU-T H.264, conocido alternativamente como MPEG-4, Parte 10, Codificación de video avanzada (AVC), o extensiones de dichos estándares. Las técnicas de esta divulgación, sin embargo, no se limitan a ningún estándar de codificación en
35 particular. Otros ejemplos de estándares de compresión de video incluyen MPEG-2 y ITU-T H.263.

Aunque no se muestra en la figura 1, en algunos aspectos, el codificador de video 20 y el decodificador de video 30 pueden cada uno estar integrados con un codificador y decodificador de audio, y pueden incluir unidades MUX-DEMUX apropiadas, u otro hardware y software, para manejar la codificación de audio y video en un flujo de datos común o
40 flujos de datos separados. Si corresponde, en algunos ejemplos, las unidades MUX-DEMUX pueden cumplir con el protocolo multiplexor ITU H.223, u otros protocolos tales como el protocolo de datagramas de usuario (UDP).

El codificador de video 20 y el decodificador de video 30 pueden implementarse cada uno como una variedad de circuitos de codificador adecuados, tales como uno o más microprocesadores, procesadores de señal digital (DSP),
45 circuitos integrados de aplicación específica (Application Specific Integrated Circuits, ASIC), matrices de puertas programables de campo (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 informáticamente no transitorio adecuado y ejecutar las instrucciones en un hardware que usa uno o más procesadores para realizar las técnicas de esta divulgación. Cada uno de entre el codificador de
50 video 20 y el decodificador de video 30 puede ser incluido en uno o más codificadores o decodificadores, pudiéndose integrar cualquiera de los cuales como parte de un codificador/decodificador combinado (CODEC) en un respectivo dispositivo.

El JCT-VC está trabajando en el desarrollo del estándar HEVC. Los esfuerzos de estandarización del HEVC se basan
55 en un modelo en evolución de un dispositivo de codificación de video denominado modelo de prueba del HEVC (HM). El HM asume varias capacidades adicionales de dispositivos de codificación de video en relación con dispositivos existentes según, por ejemplo, ITU-T H.264/AVC. Por ejemplo, mientras que el H.264 proporciona nueve modos de intra codificación-predicción, el HM puede proporcionar hasta treinta y tres modos de intra codificación-predicción.

60 En general, el modelo de trabajo del HM describe que una trama de video o imagen se puede dividir en una secuencia de bloques de árbol o unidades de codificación más grandes (Largest Coding Units, LCU) que incluyen muestras de luma y croma. Un bloque de árbol tiene un propósito similar a un macro-bloque del estándar H.264. Un segmento incluye varios bloques de árbol consecutivos en orden de codificación. Una trama de video o imagen puede dividirse

en uno o más segmentos. Cada bloque de árbol se puede dividir en unidades de codificación (CU) de acuerdo con un árbol cuaternario. Por ejemplo, un bloque de árbol como nodo raíz del árbol cuaternario, puede dividirse en cuatro nodos hijo, y cada nodo hijo puede a su vez ser un nodo padre y dividirse en otros cuatro nodos hijo. Un nodo hijo final no dividido, como nodo hoja del árbol cuaternario, comprende un nodo de codificación, es decir, un bloque de video
5 codificado. Los datos de sintaxis asociados con un flujo de bits codificado pueden definir un número máximo de veces que un bloque de árbol puede dividirse, y también puede definir un tamaño mínimo de los nodos de codificación.

Una unidad de codificación incluye un nodo de codificación y unidades de predicción (PU) y unidades de transformación (TU) asociadas con el nodo de codificación. Un tamaño de la unidad de codificación corresponde a un
10 tamaño del nodo de codificación y debe tener forma cuadrada. El tamaño de la unidad de codificación puede variar desde 8x8 píxeles hasta el tamaño del bloque de árbol con un máximo de 64x64 píxeles o más. Cada unidad de codificación puede contener una o más unidades de predicción y una o más unidades de transformación. Los datos de sintaxis asociados con una unidad de codificación pueden describir, por ejemplo, la división de la unidad de codificación en una o más unidades de predicción. Los modos de división pueden diferir entre si la unidad de
15 codificación es codificada en modo directo de salto, es codificada en modo intra-predicción o es codificada en modo inter-predicción. Las unidades de predicción se pueden dividir para que no tengan forma cuadrada. Los datos de sintaxis asociados con una unidad de codificación también pueden describir, por ejemplo, la división de la unidad de codificación en una o más unidades de transformación de acuerdo con un árbol cuaternario. Una unidad de transformación puede ser de forma cuadrada o no cuadrada.

20 El estándar HEVC permite transformaciones de acuerdo con las unidades de transformación, que pueden ser diferentes para distintas unidades de codificación. Las unidades de transformación suelen tener un tamaño basado en el tamaño de las unidades de predicción dentro de una determinada unidad de codificación definida para una unidad de codificación más grande dividida, aunque este no siempre sea el caso. Las unidades de transformación son
25 típicamente del mismo tamaño o más pequeñas que las unidades de predicción. En algunos ejemplos, las muestras residuales correspondientes a una unidad de codificación pueden subdividirse en unidades más pequeñas utilizando una estructura de árbol cuaternario conocida como "árbol cuaternario residual" (Residual Quad Tree, RQT). Los nodos hoja del árbol cuaternario residual pueden denominarse unidades de transformación (TU). Los valores de diferencia de píxeles asociados con las unidades de transformación pueden ser transformados para producir coeficientes de
30 transformación, que pueden ser cuantificados.

En general, una unidad de predicción incluye datos relacionados con el proceso de predicción. Por ejemplo, cuando la unidad de predicción es codificada en modo intra, la unidad de predicción puede incluir datos que describen un modo de intra-predicción para la unidad de predicción. Como otro ejemplo, cuando la unidad de predicción es
35 codificada en modo inter, la unidad de predicción puede incluir datos que definen un vector de movimiento para la unidad de predicción. Los datos que definen el vector de movimiento para una unidad de predicción pueden describir, por ejemplo, un componente horizontal del vector de movimiento, un componente vertical del vector de movimiento, una resolución para el vector de movimiento (por ejemplo, una precisión de un cuarto de píxel o una precisión de un octavo de píxel), una imagen de referencia a la que apunta el vector de movimiento, y/o una lista de imágenes de
40 referencia (por ejemplo, Lista 0, Lista 1 o Lista C) para el vector de movimiento.

En general, se usa una unidad de transformación para los procesos de transformación y cuantificación. Una determinada unidad de codificación que tenga una o más unidades de predicción también puede incluir una o más unidades de transformación. Después de la predicción, el codificador de video 20 puede calcular valores residuales
45 correspondientes a la unidad de predicción. Los valores residuales comprenden valores de diferencia de píxeles que se pueden transformar en coeficientes de transformación, cuantificar y explorar utilizando las unidades de transformación para producir coeficientes de transformación serializados para la codificación de entropía. El término "bloque de video" en esta divulgación puede referirse a un nodo de codificación de una unidad de codificación, o a un bloque de coeficientes de transformación. Uno o más bloques de coeficientes de transformación pueden definir una
50 unidad de transformación. En algunos casos específicos, esta divulgación también puede usar el término "bloque de video" para referirse a un bloque de árbol, es decir, una unidad de codificación más grande o una unidad de codificación, que incluye un nodo de codificación y unidades de predicción y unidades de transformación.

Una secuencia de vídeo incluye típicamente una serie de tramas de vídeo o imágenes. Un grupo de imágenes (GOP)
55 generalmente comprende una serie de una o más de las imágenes de video. Un grupo de imágenes puede incluir datos de sintaxis en un encabezado del grupo de imágenes, un encabezado de una o más de las imágenes, o en otro lugar, que describe una serie de imágenes incluidas en el grupo de imágenes. Cada segmento (slice) de una imagen puede incluir datos de sintaxis del segmento que describen un modo de codificación para el respectivo segmento. El codificador de video 20 generalmente opera en bloques de video dentro de segmentos de video individuales con el fin
60 de codificar los datos de video. Un bloque de video puede incluir una o más unidades de transformación o unidades de predicción que corresponden a un nodo de codificación dentro de una unidad de codificación. Los bloques de video pueden tener tamaños fijos o variables, y pueden diferir en tamaño según un estándar de codificación específico.

Como un ejemplo, el HM admite la predicción en diversos tamaños de unidad de predicción. Suponiendo que el tamaño de una unidad de codificación particular es $2N \times 2N$, el HM admite la intra-predicción en tamaños de unidad de predicción de $2N \times 2N$ o $N \times N$, y la inter-predicción en tamaños de unidad de predicción simétricos de $2N \times 2N$, $2N \times N$, $N \times 2N$ o $N \times N$. El HM también admite la división asimétrica para la inter-predicción en tamaños de unidad de predicción de $2N \times nU$, $2N \times nD$, $nL \times 2N$ y $nR \times 2N$. En la división asimétrica, no se divide una dirección de una unidad de codificación, mientras que la otra dirección se divide en 25% y 75%. La parte de la unidad de codificación correspondiente a la división del 25% se indica con una "n" seguida de una indicación de "Arriba" ("Up"), "Abajo" ("Down"), "Izquierda" ("Left") o "Derecha" ("Right"). Así, por ejemplo, "2NxnU" se refiere a una unidad de codificación de $2N \times 2N$ que está dividida horizontalmente con una unidad de predicción $2N \times 0,5N$ en la parte superior y una unidad de predicción $2N \times 1,5N$ en la parte inferior.

En esta divulgación, " $N \times N$ " y " N por N " se pueden usar indistintamente para referirse a las dimensiones en píxeles de un bloque de video en términos de dimensiones verticales y horizontales, por ejemplo, 16×16 píxeles o 16 por 16 píxeles. En general, un bloque de 16×16 tendrá 16 píxeles en una dirección vertical ($y = 16$) y 16 píxeles en una dirección horizontal ($x = 16$). Del mismo modo, un bloque $N \times N$ generalmente tiene N píxeles en una dirección vertical y N píxeles en una dirección horizontal, en el que N representa un valor entero no negativo. Los píxeles en un bloque se pueden organizar en filas y columnas. Además, los bloques no deben tener necesariamente el mismo número de píxeles en la dirección horizontal que en la dirección vertical. Por ejemplo, hay bloques que pueden comprender $N \times M$ píxeles, en el que M no es necesariamente igual a N .

Después de la codificación intra-predictiva o inter-predictiva usando las unidades de predicción de una unidad de codificación, el codificador de video 20 puede calcular datos residuales para las unidades de transformación de la unidad de codificación. Las unidades de predicción pueden comprender datos de píxeles en el dominio espacial (también denominado dominio de píxeles) y las unidades de transformación pueden comprender coeficientes en el dominio de transformación después de la aplicación de una transformación a los datos de video residuales, por ejemplo, una transformada de coseno discreta (DCT), una transformada entera, una transformada ondícula, o una transformada conceptualmente similar. Los datos residuales pueden corresponder a diferencias de píxeles entre píxeles de la imagen sin codificar y valores de predicción correspondientes a las unidades de predicción. El codificador de video 20 puede formar las unidades de transformación a partir de uno o más bloques de coeficientes de transformación. Las unidades de transformación pueden incluir datos residuales para la unidad de codificación. El codificador de video 20 puede entonces transformar las unidades de transformación para producir coeficientes de transformación para la unidad de codificación.

Después de cualesquiera transformaciones para producir coeficientes de transformación, el codificador de video 20 puede realizar la cuantificación de los coeficientes de transformación. La cuantificación generalmente se refiere a un proceso en el cual se cuantifican los coeficientes de transformación para reducir posiblemente la cantidad de datos utilizados para representar los coeficientes, proporcionando una compresión adicional. El proceso de cuantificación puede reducir la profundidad de bits asociada con algunos o todos los coeficientes. Por ejemplo, un valor de n bits puede ser redondeado hacia abajo a un valor de m bits durante la cuantificación, en el que n es mayor que m .

En algunos ejemplos, el codificador de video 20 puede utilizar un orden de exploración predefinido para explorar los coeficientes de transformación cuantificados para producir un vector serializado que puede ser codificado por entropía. En otros ejemplos, el codificador de video 20 puede realizar una exploración adaptativa. Las figuras 2A-2D ilustran diferentes órdenes de exploración de ejemplo. También se pueden usar otros órdenes de exploración definidos, u órdenes de exploración adaptativos (cambiantes). La figura 2A ilustra un orden de exploración en zig-zag, 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 en diagonal. También se pueden definir y utilizar combinaciones de estos órdenes de exploración. En algunos ejemplos, las técnicas de esta divulgación pueden ser aplicables específicamente durante la codificación de un denominado mapa de significancia en el proceso de codificación de video.

Se pueden definir uno o más elementos de sintaxis para indicar una posición de un último coeficiente significativo (es decir, un coeficiente distinto de cero), que puede depender del orden de exploración asociado con un bloque de coeficientes. Por ejemplo, un elemento de sintaxis puede definir una posición de columna de un último coeficiente significativo dentro de un bloque de valores de coeficientes y otro elemento de sintaxis puede definir una posición de fila del último coeficiente significativo dentro de un bloque de valores de coeficientes.

La figura 3 ilustra un ejemplo de un mapa de significancia relativo a un bloque de valores de coeficiente. El mapa de significancia se muestra a la derecha, en el que unos indicadores de un bit identifican los coeficientes del bloque de video de la izquierda que son significativos, es decir, que son distintos de cero. En un ejemplo, dado un conjunto de coeficientes significativos (por ejemplo, definidos por un mapa de significancia) y un orden de exploración, se puede definir una posición de un último coeficiente significativo. En el estándar HEVC emergente, los coeficientes de transformación se pueden agrupar en un fragmento. El fragmento puede comprender una unidad de transformación completa, o en algunos casos, las unidades de transformación se pueden subdividir en fragmentos más pequeños. El

mapa de significancia y la información de nivel (valor absoluto y signo) son codificados para cada coeficiente en un fragmento. En un ejemplo, un fragmento consta de 16 coeficientes consecutivos en un orden de exploración inverso (por ejemplo, diagonal, horizontal, o vertical) para una unidad de transformación 4×4 y una unidad de transformación 8×8. Para unidades de transformación 16×16 y 32×32, los coeficientes dentro de un sub-bloque 4×4 son tratados

- 5 como un fragmento. Los elementos de sintaxis son codificados y señalizados para representar la información de nivel de los coeficientes dentro de un fragmento. En un ejemplo, todos los símbolos son codificados en un orden de exploración inverso. Las técnicas de esta divulgación pueden mejorar la codificación de un elemento de sintaxis utilizado para definir esta posición del último coeficiente significativo de un bloque de coeficientes.
- 10 Como un ejemplo, las técnicas de la presente divulgación se pueden utilizar para codificar la posición del último coeficiente significativo de un bloque de coeficientes (por ejemplo, una unidad de transformación o un fragmento de una unidad de transformación). Después de codificar la posición del último coeficiente significativo, se puede codificar la información de nivel y de signo. La codificación de la información de nivel y de signo puede procesarse de acuerdo con un enfoque de cinco pasos codificando los siguientes símbolos en orden de exploración inverso (por ejemplo, para
- 15 una unidad de transformación o un fragmento de la unidad de transformación):

significant_coeff_flag (abreviatura: sigMapFlag): este indicador puede indicar el significado de cada coeficiente en un fragmento (chunk). Un coeficiente con un valor de uno o mayor se considera significativo.

- 20 coeff_abs_level_greater1_flag (abreviatura: gr1Flag): este indicador puede indicar si el valor absoluto del coeficiente es mayor que uno para los coeficientes distintos de cero (es decir, los coeficientes con sigMapFlag igual a 1).

coeff_abs_level_greater2_flag (abreviatura: gr2Flag): este indicador puede indicar si el valor absoluto del coeficiente es mayor que dos para los coeficientes con un valor absoluto mayor que uno (es decir, los coeficientes con gr1Flag

25 igual a 1).

coeff_sign_flag (abreviatura: signFlag): este indicador puede indicar información de signo para los coeficientes distintos de cero. Por ejemplo, un cero para este indicador indica un signo positivo, mientras que un 1 indica un signo

30 negativo.

- coeff_abs_level_remain (abreviatura: levelRem): es el valor absoluto restante de un nivel de coeficientes de transformación. Para este indicador, se codifica el valor absoluto del coeficiente $-x$ ($\text{abs}(\text{nivel})-x$) para cada coeficiente con la amplitud mayor que x el valor de x depende de los presentes de gr1Flag y gr2Flag.

- 35 De este modo, se pueden codificar los coeficientes de transformación para una unidad de transformación o un fragmento de una unidad de transformación. En cualquier caso, las técnicas de esta divulgación, que se refieren a la codificación de un elemento de sintaxis que se usa para definir la posición del último coeficiente significativo de un bloque de coeficientes, también se pueden usar con otros tipos de técnicas para codificar finalmente la información de nivel y signo de los coeficientes de transformación. El enfoque de cinco pasos para codificar la información de
- 40 significancia, nivel y signo es solo una técnica de ejemplo que se puede usar después de la codificación de la posición del último coeficiente significativo de un bloque, según se establece en esta divulgación.

Después de explorar los coeficientes de transformación cuantificados para formar un vector unidimensional, el codificador de video 20 puede codificar por entropía el vector unidimensional, por ejemplo, de acuerdo con la

45 codificación de longitud variable adaptativa al contexto (CAVLC), la codificación aritmética binaria adaptativa al contexto (CABAC), la codificación aritmética binaria adaptativa al contexto basada en sintaxis (SBAC), codificación de entropía de división de intervalos de probabilidad (PIPE) u otra metodología de codificación de entropía. El codificador de video 20 también puede codificar por entropía elementos de sintaxis asociados con los datos de video codificados para su uso por el decodificador de video 30 en la decodificación de los datos de video. Las técnicas de codificación

50 de entropía de esta divulgación se describen específicamente como aplicables a CABAC, aunque las técnicas también pueden ser aplicables a otras técnicas de codificación de entropía tales como CAVLC, SBAC, PIPE u otras técnicas.

Para realizar CABAC, el codificador de video 20 puede asignar un contexto dentro de un modelo de contexto a un símbolo a transmitir. El contexto puede estar relacionado con, por ejemplo, si valores vecinos del símbolo son distintos

55 de cero o no. Para realizar CAVLC, el codificador de video 20 puede seleccionar un código de longitud variable para un símbolo a transmitir. En la codificación de longitud variable (VLC) se pueden construir palabras de código de tal manera que los códigos relativamente más cortos correspondan a símbolos más probables, mientras que los códigos más largos correspondan a símbolos menos probables. De esta manera, el uso de VLC puede conseguir un ahorro de bits con respecto a, por ejemplo, el uso de palabras de código de igual longitud para cada símbolo a transmitir. La

60 determinación de la probabilidad se puede basar en un contexto asignado al símbolo.

En general, la codificación de símbolos de datos utilizando CABAC puede implicar una o más de las siguientes etapas:

(1) Binarización: si un símbolo a codificar no tiene valor binario, es mapeado con una secuencia de los denominados "bins". Cada valor binario (bin) puede tener un valor de "0" o "1".

(2) Asignación de contexto: cada bin (en modo regular) es asignado a un contexto. Un modelo de contexto determina cómo se calcula un contexto para un bin determinado en función de la información disponible para el bin, tal como valores de símbolos codificados previamente o número de bin.

(3) Codificación de bins: los bins se codifican con un codificador aritmético. Para codificar un bin, el codificador aritmético requiere como entrada una probabilidad del valor del bin, es decir, una probabilidad de que el valor del bin sea igual a "0", y una probabilidad de que el valor del valor binario (bin) sea igual a "1". La probabilidad (estimada) de cada contexto es representada por un valor entero denominado "estado de contexto". Cada contexto tiene un estado y, por lo tanto, el estado (es decir, la probabilidad estimada) es el mismo para los bins asignados a un contexto, y es diferente entre contextos.

(4) Actualización de estados: la probabilidad (estado) para un contexto seleccionado es actualizada en función del valor codificado real del bin (por ejemplo, si el valor del bin era "1", la probabilidad de "1s" es mayor).

Se debe observar que la codificación de entropía de división de intervalos de probabilidad (PIPE) utiliza principios similares a los de la codificación aritmética, y puede utilizar técnicas similares a las de esta divulgación, que se describen principalmente con respecto a CABAC. Las técnicas de esta divulgación, sin embargo, pueden usarse con CABAC, PIPE u otras metodologías de codificación de entropía que utilizan técnicas de binarización.

Una técnica adoptada recientemente en HM4.0 se describe en V. Seregin, I.-K Kim, "Binarisation modification for last position coding", JCTVC-F375, 6thJCT-VC Meeting, Torino, IT, Julio, 2011 ("Seregin" en este documento). La técnica adoptada en HM4.0 reduce los contextos utilizados en la codificación de la última posición para CABAC introduciendo códigos de longitud fija con un modo de bypass. El modo de bypass significa que no hay un procedimiento de modelado de contexto y que cada símbolo es codificado con el mismo estado de probabilidad. Aumentar el número de bins codificados en modo de bypass mientras se reducen los bins en modo regular puede ayudar a acelerar y paralelizar el códec.

En la técnica adoptada en HM4.0, la magnitud máxima posible del componente de última posición, max_length, se divide por igual en dos mitades. La primera mitad es codificada con un código unario truncado y la segunda mitad es codificada con códigos de longitud fija (siendo el número de bins igual a $\log_2(\text{max_length}/2)$). En el peor de los casos, el número de bins que utiliza el modelado de contexto es igual a max_length/2. La tabla 1 muestra la binarización para una unidad de transformación de 32x32 en HM4.0.

Tabla 1. Binarización para una unidad de transformación de 32x32 en HM4.0, en el que X significa 1 o 0.

Magnitud del componente de última posición	Unario truncado (modelo de contexto)	Binario fijo (bypass)
0	1	-
1	01	-
2	001	-
3	0001	-
4	00001	-
5	000001	-
6	0000001	-
7	00000001	-
8	000000001	-
9	0000000001	-
10	00000000001	-
11	000000000001	-
12	0000000000001	-
13	00000000000001	-
14	000000000000001	-
15	0000000000000001	-
16-31	0000000000000000	XXXX

Esta divulgación proporciona técnicas para la codificación aritmética binaria adaptativa al contexto (CABAC) de la posición de último coeficiente significativo. En un ejemplo, se puede usar una estructura de palabra de código progresiva con un número reducido de bins y códigos unarios truncados más cortos. Además, en un ejemplo,

reduciendo la longitud máxima del código unario truncado, el número de modelos de contexto para la posición de último coeficiente significativo puede reducirse en dos.

La figura 4 es un diagrama de bloques que ilustra un codificador de video de ejemplo 20 que puede implementar las técnicas descritas en esta divulgación. El codificador de video 20 puede realizar intra- e inter-codificación de bloques de video dentro de segmentos de video. La intra-codificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial en el video dentro de una trama o imagen de video determinado. La inter-codificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal en el video dentro de tramas o imágenes adyacentes de una secuencia de video. El intra-modo (modo I) puede referirse a cualquiera de entre varios modos de compresión basados en el espacio. Los inter-modos, tal como la predicción unidireccional (modo P) o la predicción bidireccional (modo B), pueden referirse a cualquiera de entre varios modos de compresión basados en el tiempo.

En el ejemplo de la figura 4, el codificador de video 20 incluye una unidad de división 35, un módulo de predicción 41, una memoria de imágenes de referencia 64, un agregador 50, un módulo de transformación 52, una unidad de cuantificación 54 y una unidad de codificación de entropía 56. El módulo de predicción 41 incluye una unidad de estimación de movimiento 42, una unidad de compensación de movimiento 44, y un módulo de intra-predicción 46. Para la reconstrucción de bloques de video, el codificador de video 20 también incluye una unidad de cuantificación inversa 58, un módulo de transformación inversa 60 y un agregador 62. También se puede incluir un filtro de desbloqueo (no mostrado en la Figura 2) para filtrar los límites de los bloques para eliminar artefactos de bloqueo del video reconstruido. Si se desea, el filtro de desbloqueo normalmente filtraría la salida del agregador 62. También se pueden usar filtros de bucle adicionales (en bucle o post bucle) además del filtro de desbloqueo.

Según se muestra en la figura 4, el codificador de video 20 recibe datos de video, y la unidad de división 35 divide los datos en bloques de video. Esta división también puede incluir la división en segmentos, piezas de mosaico u otras unidades más grandes, así como la división de bloques de video, por ejemplo, de acuerdo con una estructura de árbol cuaternario de unidades de codificación más grandes y unidades de codificación. El codificador de video 20 ilustra generalmente los componentes que codifican bloques de video dentro de un segmento de video a codificar. El segmento puede dividirse en múltiples bloques de video (y posiblemente en conjuntos de bloques de video denominados piezas de mosaico). El módulo de predicción 41 puede seleccionar uno de entre una pluralidad de posibles modos de codificación, tal como uno de entre una pluralidad de modos de intra codificación o uno de entre una pluralidad de modos de inter codificación, para el bloque de video actual en base a resultados de error (por ejemplo, tasa de codificación y nivel de distorsión). El módulo de predicción 41 puede proporcionar el bloque intra- o inter-codificado resultante al agregador 50 para generar datos de bloque residuales y al agregador 62 para reconstruir el bloque codificado para su uso como imagen de referencia.

El módulo de inter-predicción 46 dentro del módulo de predicción 41 puede realizar una codificación intra-predictiva del bloque de video actual en relación con uno o más bloques vecinos en la misma trama o segmento que el bloque actual a codificar para proporcionar una compresión espacial. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 dentro del módulo de predicción 41 realizan la codificación inter-predictiva del bloque de video actual en relación con uno o más bloques predictivos en una o más imágenes de referencia para proporcionar una compresión temporal.

La unidad de estimación de movimiento 42 puede estar configurada para determinar el modo de predicción para un segmento de video de acuerdo con un patrón predeterminado para una secuencia de video. El patrón predeterminado puede designar segmentos de video en la secuencia como segmentos predictivos (segmentos P), segmentos bi-predictivos (segmentos B) o segmentos P y B generalizados (segmentos GPB). Un segmento P puede referenciar a una imagen secuencial anterior. Un segmento B puede referenciar a una imagen secuencial anterior o una imagen secuencial posterior. Los segmentos GPB se refieren a un caso en el que dos listas de imágenes de referencia son idénticas. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden estar muy integradas, pero se ilustran por separado para fines conceptuales. La estimación de movimiento, realizada por la unidad de estimación de movimiento 42, es el proceso de generación de vectores de movimiento, que estiman el movimiento para bloques de video. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de una unidad de predicción de un bloque de video dentro de una trama o imagen de video actual en relación con un bloque predictivo dentro de una imagen de referencia.

Un bloque predictivo es un bloque que se determina que tiene una gran correspondencia con la unidad de predicción del bloque de video a codificar en términos de diferencia de píxeles, lo que puede ser determinado por la suma de la diferencia absoluta (SAD), la suma de la diferencia de cuadrados (SSD), u otras métricas de diferencia. En algunos ejemplos, el codificador de video 20 puede calcular valores para las posiciones de píxeles sub-enteros de imágenes de referencia almacenadas en la memoria de imágenes de referencia 64. Por ejemplo, el codificador de video 20 puede interpolar valores de posiciones de un cuarto de píxel, posiciones de un octavo de píxel u otras posiciones de píxeles fraccionales de la imagen de referencia. Por lo tanto, la unidad de estimación de movimiento 42 puede realizar

una búsqueda de movimiento relativo a las posiciones de píxeles completas y las posiciones de píxeles fraccionales y generar un vector de movimiento con una precisión de píxeles fraccionales.

La unidad de estimación de movimiento 42 calcula un vector de movimiento para una unidad de predicción de un
 5 bloque de video en un segmento inter-codificado comparando la posición de la unidad de predicción con la posición
 de un bloque predictivo de una imagen de referencia. La imagen de referencia se puede seleccionar de una primera
 lista de imágenes de referencia (Lista 0) o una segunda lista de imágenes de referencia (Lista 1), identificando cada
 una de las cuales una o más imágenes de referencia almacenadas en la memoria de imágenes de referencia 64. La
 unidad de estimación de movimiento 42 envía el vector de movimiento calculado a la unidad de codificación de entropía
 10 56 y a la unidad de compensación de movimiento 44.

La compensación de movimiento, realizada por la unidad de compensación de movimiento 44, puede implicar
 recuperar o generar el bloque predictivo basándose en el vector de movimiento determinado por la estimación de
 movimiento, posiblemente realizando interpolaciones con una precisión de sub-píxeles. Al recibir el vector de
 15 movimiento para la unidad de predicción del bloque de video actual, la unidad de compensación de movimiento 44
 puede localizar el bloque predictivo al que apunta el vector de movimiento en una de las listas de imágenes de
 referencia. El codificador de video 20 forma un bloque de video residual restando valores de píxeles del bloque
 predictivo a los valores de píxeles del bloque de video actual que se está codificando, formando con ello unos valores
 de diferencia de píxeles. Los valores de diferencia de píxeles conforman datos residuales para el bloque, y pueden
 20 incluir componentes de diferencia tanto de luma como de croma. El agregador 50 representa el componente o
 componentes que realizan esta operación de resta. La unidad de compensación de movimiento 44 también puede
 generar elementos de sintaxis asociados con los bloques de video y el segmento de video para su uso por parte del
 decodificador de video 30 en la decodificación de los bloques de video del segmento de video.

El módulo de intra-predicción 46 puede intra-pronosticar un bloque actual, como alternativa a la inter-predicción
 realizada por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44, según se ha
 descrito anteriormente. En particular, el módulo de intra-predicción 46 puede determinar un modo de intra-predicción
 a usar para codificar un bloque actual. En algunos ejemplos, el módulo de intra-predicción 46 puede codificar un bloque
 actual utilizando varios modos de intra-predicción, por ejemplo, durante pasos de codificación independientes, y el
 30 módulo de intra-predicción 46 (o unidad de selección de modo 40 en algunos ejemplos) puede seleccionar de entre
 los modos probados un modo de intra-predicción a utilizar apropiado. Por ejemplo, el módulo de intra-predicción 46
 puede calcular valores de tasa-distorsión usando un análisis de tasa-distorsión para los diversos modos de intra-
 predicción probados, y seleccionar el modo de intra-predicción que tiene las mejores características de tasa-distorsión
 de entre los modos probados. El análisis de tasa-distorsión generalmente determina una cantidad de distorsión (o
 35 error) entre un bloque codificado y un bloque original no codificado que fue codificado para producir el bloque
 codificado, así como una tasa de bits (es decir, una cantidad de bits) utilizada para producir el bloque codificado. El
 módulo de intra-predicción 46 puede calcular relaciones de las distorsiones y tasas de los diversos bloques codificados
 para determinar qué modo de intra-predicción presenta el mejor valor de tasa-distorsión para el bloque.

En cualquier caso, después de seleccionar un modo de intra-predicción para un bloque, el módulo de intra-predicción
 46 puede proporcionar información indicativa del modo de intra-predicción seleccionado para el bloque a la unidad de
 codificación de entropía 56. La unidad de codificación de entropía 56 puede codificar la información que indica el modo
 de intra-predicción seleccionado de acuerdo con las técnicas de esta divulgación. El codificador de video 20 puede
 45 incluir en los datos transmitidos de configuración del flujo de bits, que pueden incluir una pluralidad de tablas de índices
 de modos de intra-predicción y una pluralidad de tablas de índices de modos de intra-predicción modificados (también
 conocidas como tablas de mapeo de palabras de código), definiciones de contextos de codificación para varios
 bloques, e indicaciones de un modo de intra-predicción más probable, una tabla de índices de modos de intra-
 predicción y una tabla de índices de modos de intra-predicción modificados a usar en cada uno de los contextos.

Después de que el módulo de predicción 41 genere el bloque predictivo para el bloque de video actual a través de
 inter-predicción o intra-predicción, el codificador de video 20 forma un bloque de video residual restando el bloque
 predictivo al bloque de video actual. Los datos de video residuales en el bloque residual pueden ser incluidos en una
 o más unidades de transformación y aplicados al módulo de transformación 52. El módulo de transformación 52
 50 transforma los datos de video residuales en coeficientes de transformación residuales utilizando una transformada, tal
 como una transformada de coseno discreta (DCT) o una transformación conceptualmente similar. El módulo de
 transformación 52 puede convertir los datos de video residuales desde un dominio de píxeles a un dominio de
 55 transformación, tal como un dominio de frecuencia.

El módulo de transformación 52 puede enviar los coeficientes de transformación resultantes a la unidad de
 60 cuantificación 54. La unidad de cuantificación 54 cuantifica los coeficientes de transformación para reducir aún más la
 tasa de bits. El proceso de cuantificación puede reducir la profundidad de bits asociada con algunos o todos los
 coeficientes. El grado de cuantificación puede modificarse ajustando un parámetro de cuantificación. En algunos
 ejemplos, la unidad de cuantificación 54 puede entonces realizar una exploración de la matriz que incluye los

coeficientes de transformación cuantificados. Alternativamente, la unidad de codificación de entropía 56 puede realizar la exploración. La unidad de cuantificación inversa 58 y el módulo de transformación inversa 60 aplican cuantificación inversa y transformación inversa, respectivamente, para reconstruir el bloque residual en el dominio de píxeles para su uso posterior como un bloque de referencia de una imagen de referencia. La unidad de compensación de movimiento 44 puede calcular un bloque de referencia agregando el bloque residual a un bloque predictivo de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. La unidad de compensación de movimiento 44 también puede aplicar uno o más filtros de interpolación al bloque residual reconstruido para calcular los valores de píxeles sub-enteros para usarlos en la estimación de movimiento. El agregador 62 agrega el bloque residual reconstruido al bloque de predicción de movimiento compensado producido por la unidad de compensación de movimiento 44 para producir un bloque de referencia para su almacenamiento en la memoria de imágenes de referencia 64. El bloque de referencia puede ser utilizado por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 como un bloque de referencia para inter-predecir un bloque en una trama de video o imagen posterior.

15 Después de la cuantificación, la unidad de codificación de entropía 56 codifica por entropía los coeficientes de transformación cuantificados. Por ejemplo, la unidad de codificación de entropía 56 puede realizar codificación de longitud variable adaptativa al contexto (CAVLC), codificación aritmética binaria adaptativa al contexto (CABAC), codificación aritmética binaria adaptativa al contexto basada en sintaxis (SBAC), codificación de entropía de división de intervalos de probabilidad (PIPE) u otra metodología o técnica de codificación de entropía. Después de la codificación de entropía realizada por la unidad de codificación de entropía 56, puede transmitirse el flujo de bits codificado al decodificador de video 30, o archivarse para su posterior transmisión o recuperación por parte del decodificador de video 30. La unidad de codificación de entropía 56 también puede codificar por entropía los vectores de movimiento y los otros elementos de sintaxis para el segmento de video actual que está siendo codificado.

25 En un ejemplo, la unidad de codificación de entropía 56 puede codificar la posición de un último coeficiente significativo usando la técnica adoptada en HM4.0 descrita anteriormente. En otros ejemplos, la unidad de codificación de entropía 56 puede codificar la posición de un último coeficiente significativo utilizando las técnicas que pueden proporcionar una codificación mejorada. En particular, la unidad de codificación de entropía 56 puede utilizar un esquema de codificación progresiva de la última posición para varios tamaños posibles de unidad de transformación.

30 En un ejemplo, una palabra de código para la posición de un último coeficiente significativo puede incluir un prefijo de código unario truncado seguido de un sufijo de código de longitud fija. En un ejemplo, cada magnitud de la última posición puede usar la misma binarización para todos los tamaños posibles de unidad de transformación, excepto cuando la última posición es igual al tamaño de la unidad de transformación menos 1. Esta excepción se debe a las propiedades de la codificación unaria truncada. En un ejemplo, se puede especificar la posición de un último coeficiente significativo dentro de un coeficiente de transformación rectangular especificando un valor de coordenada-x y un valor de coordenada-y. En otro ejemplo, un bloque de coeficientes de transformación puede tener la forma de un vector 1xN y la posición del último coeficiente significativo dentro del vector puede ser especificada por un valor de posición de señal.

40 En un ejemplo, T puede definir el tamaño de una unidad de transformación. Como se ha descrito en detalle anteriormente, una unidad de transformación puede tener forma cuadrada o no cuadrada. Por lo tanto, T puede referirse al número de filas o columnas de una unidad de transformación de dos dimensiones o a la longitud de un vector. En un ejemplo en el que un esquema de codificación unaria truncada proporciona un número de bits iguales a cero seguidos por un bit igual a uno, el número de ceros de un prefijo de código unario truncado que codifica la posición de un último coeficiente significativo puede definirse de acuerdo con $N = \{0, \dots, 2\log_2(T)-1\}$. Cabe señalar que en un ejemplo en el que un esquema de codificación unaria truncada proporciona un número de bits iguales a uno seguidos de un bit igual a cero, $N = \{0, \dots, 2\log_2(T)-1\}$ también puede definir el número de unos. En cada una de estas alternativas de codificación unaria truncada, $2\log_2(T)-1$ puede definir la longitud máxima de un prefijo unario truncado para una unidad de transformación de tamaño T. Por ejemplo, para una unidad de transformación en la que T es igual a 32, la longitud máxima del prefijo unario truncado es igual a nueve y en el que T es igual a 16, la longitud máxima del prefijo unario truncado es igual a siete.

Para un código unario truncado, valor n, un sufijo de código de longitud fija puede incluir los siguientes b bits de código binario de longitud fija con un valor definido como sigue: $f_value = \{0, \dots, 2^b-1\}$, en el que $b = \max(0, n/2-1)$. Por lo tanto, la magnitud de la última posición, last_pos, se puede derivar de n y f_value de la siguiente manera:

$$last_pos = \begin{cases} n, & si\ n < 2 \\ 2^{\frac{n}{2}-1} \times (2 + \text{mod}(n,2)) + f_value, & si\ n \geq 2 \end{cases} \quad (1)$$

60 en la que mod(.) representa la operación modular y f_value representa el valor del código de longitud fija.

La Tabla 2 muestra un ejemplo de binarización de una posición de un último coeficiente significativo de acuerdo con las definiciones proporcionadas según la Ecuación 1 para una unidad de transformación de 32x32. La segunda columna de la Tabla 2 proporciona valores correspondientes a prefijos unarios truncados para los valores posibles de la posición de un último coeficiente significativo dentro de una unidad de transformación de tamaño T definido por la longitud máxima de prefijo unario truncado de $2\log_2(T)-1$. La tercera columna de la Tabla 2 proporciona un correspondiente sufijo de longitud fija para cada prefijo unario truncado. Por motivos de brevedad, la Tabla 2 incluye valores X que indican un valor de bit igual a uno o cero. Cabe señalar que los valores X se mapean de forma única con cada valor que comparte un prefijo unario truncado de acuerdo con un código de longitud fija. La magnitud del componente de última posición en la Tabla 2 puede corresponder a un valor de coordenada-x y/o a un valor de coordenada-y.

Tabla 2. Binarización para una unidad de transformación de tamaño 32x32, en la que X significa 1 o 0.

Magnitud del componente de última posición	Unario truncado (modelo de contexto)	Binario fijo (bypass)	f_value
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

15

Las Tablas 3 y 4 muestran una comparación de la longitud máxima de cadenas de bits para el esquema de binarización de ejemplo descrito con respecto a la Tabla 1 y el esquema de binarización de ejemplo descrito con respecto a la Tabla 2. Según se muestra en la Tabla 3, el prefijo de código unario puede tener una longitud máxima de 16 bins para una unidad de transformación de 32x32 en el ejemplo descrito con respecto a la Tabla 1. Mientras que el prefijo de código unario puede tener una longitud máxima de 16 bins para una unidad de transformación de 32x32 en el ejemplo descrito con respecto a la Tabla 2. Además, Según se muestra en la Tabla 4, la longitud total de la basada en el prefijo unario truncado y el sufijo de longitud fija, el número máximo de bins para el ejemplo descrito con respecto a la Tabla 2 puede ser 24 en el peor de los casos, es decir, cuando la última posición se encuentra al final de una unidad de transformación de 32x32, mientras que el peor de los casos para el ejemplo descrito con respecto a la Tabla 1 puede ser 40.

Tabla 3. Longitud máxima del código unario truncado.

	Tabla 1	Tabla 2
Unidad de transformación 4x4	3	3
Unidad de transformación 8x8	4	5
Unidad de transformación 16x16	8	7
Unidad de transformación 32x32	16	9
Total(Luma)	31	24
Total(croma)	15	15

30

Tabla 4. Longitud máxima de los bins en un componente de última posición.

	Tabla 1	Tabla 2
Unidad de transformación 4x4	3	3
Unidad de transformación 8x8	6	6
Unidad de transformación 16x16	11	9
Unidad de transformación 32x32	20	12

35

En otro ejemplo, se puede definir un esquema de codificación unaria truncada que proporciona un número de bits iguales a cero seguidos por un bit igual a uno o un número de bits iguales a uno seguidos por un bit igual a uno un prefijo de código unario truncado que codifica la posición de un último coeficiente significativo, de acuerdo con $N =$

{0,...,log₂(T)+1}. En cada uno de estos esquemas de codificación unaria truncada, log₂(T)+1 puede definir la longitud máxima de un prefijo unario truncado para una unidad de transformación de tamaño T. Por ejemplo, para una unidad de transformación en la que T es igual a 32, la longitud máxima del prefijo unario truncado es igual a seis y en la que T es igual a 8, la longitud máxima del prefijo unario truncado es igual a cinco.

5 Para un código unario truncado, el valor n, un sufijo de código de longitud fija puede incluir los siguientes b bits de código binario de longitud fija con un valor definido de la siguiente manera: f_value = {0,..., 2^b-1}, en el que b=n-2. Por lo tanto, la magnitud de la última posición, last_pos, se puede derivar de n y f_value de la siguiente manera:

$$last_pos = \begin{cases} n, & \text{si } n < 4 \\ 2^{n-2} + f_value, & \text{si } n \geq 4 \end{cases} \quad (2)$$

10

en la que f_value representa el valor del código de longitud fija.

15 La Tabla 5 muestra un ejemplo de binarización de una posición de un último coeficiente significativo de acuerdo con las definiciones proporcionadas según la Ecuación 2 para una unidad de transformación de 32x32. La segunda columna de la Tabla 5 proporciona correspondientes valores de prefijo unario truncado para posibles valores de la posición de un último coeficiente significativo dentro de una unidad de transformación de tamaño T definido por la longitud máxima del prefijo unario truncado de log₂(T)+1. La tercera columna de la Tabla 5 proporciona un correspondiente sufijo de longitud fija para cada prefijo unario truncado. Por motivos de brevedad, la Tabla 2 incluye valores X que indican un valor de bit igual a uno o cero. Cabe señalar que los valores X se mapean de forma única con cada valor que comparte un prefijo unario truncado de acuerdo con un código de longitud fija. La magnitud del componente de última posición en la Tabla 5 puede corresponder a un valor de coordenada-x y/o a un valor de coordenada-y.

25 **Tabla 5. Ejemplo de binarización para unidad de transformación 32x32, en el que X significa 1 o 0.**

Magnitud del componente de última posición	Unario truncado (modelo de contexto)	Binario fijo (bypass)	f_value
0	1	-	0
1	01	-	0
2	001	-	0
3	0001	-	0
4-7	00001	XX	0-3
8-15	000001	XXX	0-7
16-31	000000	XXXX	0-15

30 En otro ejemplo, se puede definir un esquema de codificación unaria truncada que proporciona un número de bits iguales a cero seguidos por un bit igual a uno o un número de bits iguales a uno seguidos por un bit igual a uno un prefijo de código unario truncado que codifica la posición de un último coeficiente significativo, de acuerdo con N = {0,...,log₂(T)}. En cada uno de estos esquemas de codificación unaria truncada, log₂(T) puede definir la longitud máxima de un prefijo unario truncado para una unidad de transformación de tamaño T. Por ejemplo, para una unidad de transformación en el que T es igual a 32, la longitud máxima del prefijo unario truncado es igual a cinco y en el que T es igual a 8, la longitud máxima del prefijo unario truncado es igual a cinco.

40 Para un código unario truncado, valor n, un sufijo de código de longitud fija puede incluir los siguientes b bits de código binario de longitud fija con un valor definido de la siguiente manera: f_value = {0,..., 2^b-1}, en el que b=n-1. Por lo tanto, la magnitud de la última posición, last_pos, puede derivarse de n y f_value de la siguiente manera:

$$last_pos = \begin{cases} n, & \text{si } n < 2 \\ 2^{n-1} + f_value, & \text{si } n \geq 2 \end{cases} \quad (3)$$

en la que f_value representa el valor del código de longitud fija.

45 La Tabla 6 muestra un ejemplo de binarización de una posición de un último coeficiente significativo de acuerdo con las definiciones proporcionadas según la Ecuación 3 para una unidad de transformación de 32x32. La segunda columna de la Tabla 6 proporciona correspondientes valores de prefijos unarios truncados para valores posibles de la

posición de un último coeficiente significativo dentro de una unidad de transformación de tamaño T definido por la longitud máxima del prefijo unario truncado de $\log_2(T)$. La tercera columna de la Tabla 6 proporciona un correspondiente sufijo de longitud fija para cada prefijo unario truncado. Por motivos de brevedad, la Tabla 6 incluye valores X que indican un valor de bit igual a uno o cero. Cabe señalar que los valores X se mapean de forma única con cada valor que comparte un prefijo unario truncado de acuerdo con un código de longitud fija. La magnitud del componente de última posición en la Tabla 6 puede corresponder a un valor de coordenada-x y/o a un valor de coordenada-y.

Tabla 6. Ejemplo de binarización para unidad de transformación 32x32, en el que X significa 1 o 0.

Magnitud del componente de última posición	Unario truncado (modelo de contexto)	Binario fijo (bypass)	f_value
0	1	-	0
1	01	-	0
2-3	001	X	0-1
4-7	0001	XX	0-3
8-15	00001	XXX	0-7
16-31	00000	XXXX	0-15

Las tablas 5 y 6 muestran algunos ejemplos alternativos de uso de un prefijo unario truncado y un sufijo de longitud fija para codificar la posición de un último coeficiente significativo. Los ejemplos, que se muestran en las Tablas 5 y 6, permiten bins más cortos que el ejemplo proporcionado con respecto a la Tabla 2. Cabe señalar que en el ejemplo en el que la posición del último coeficiente significativo se determina en base a un valor de coordenada-x y a un valor de coordenada-y, puede seleccionarse independientemente cualquiera de los esquemas de binarización de ejemplo mostrados en las Tablas 1, 2, 5 y 6 para el valor de la coordenada-x y el valor de la coordenada-y. Por ejemplo, el valor de la coordenada-x puede codificarse en función del esquema de binarización descrito con respecto a la Tabla 2, mientras que el valor de la coordenada-y puede codificarse en función del esquema de binarización descrito con respecto a la Tabla 6.

Como se ha descrito anteriormente, la codificación de símbolos de datos utilizando CABAC puede implicar uno o más de las siguientes etapas de binarización y de asignación de contexto. En un ejemplo, se puede usar el modelo de contexto de valor de última posición para la codificación aritmética de las cadenas unarias truncadas, mientras que no se puede usar el modelo de contexto para la codificación aritmética de las cadenas de bits fijas (es decir, omitido). En el caso en el que se codifican las cadenas unarias truncadas utilizando modelos de contexto, se asigna un contexto a cada uno de los índices bin de una cadena de bits. Índices de bin individuales pueden compartir una asignación de contexto. El número de asignaciones de contexto es igual al número de índices bin o la longitud de una cadena unaria truncada. Por lo tanto, en los casos de los ejemplos ilustrados en las Tablas 1, 2, 5 y 6, pueden especificarse tablas de contexto asociadas de acuerdo con el esquema de binarización. La Tabla 7 ilustra una posible indexación de contexto para cada bin de diferentes tamaños de unidad de transformación para las binarizaciones de ejemplo proporcionadas anteriormente con respecto a la Tabla 2 anterior. Debe observarse que el ejemplo de indexación de contexto que se proporciona en la Tabla 7 proporciona dos contextos menos que la indexación de contexto que se proporciona en Seregin.

Tabla 7

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

Las tablas 8 a 11 ilustran algunos ejemplos de indexación de contexto de acuerdo con las siguientes reglas creadas para el modelado de contexto:

1. Primeros K bins no están compartiendo el contexto, en el que $K > 1$. K podría ser diferente para cada tamaño de unidad de transformación.
2. Un contexto solo puede asignarse a bins consecutivos. Por ejemplo, bin3 – bin5 podría usar el contexto 5. Pero no está permitido que bin3 y bin5 usen el contexto 5 y bin4 use el contexto 6.
3. Los últimos N bins, $N \geq 0$, de diferentes tamaños de unidad de transformación pueden compartir el mismo contexto.

4. El número de bins que comparten el mismo contexto aumenta con los tamaños de unidad de transformación.

Las reglas 1 – 4 anteriores pueden ser particularmente útiles para la binarización proporcionada en la Tabla 2. Sin embargo, puede ajustarse el modelado de contexto de acuerdo con el esquema de binarización implementado.

Tabla 8

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

10

Tabla 9

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

Tabla 10

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

15

Tabla 11

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

La figura 5 es un diagrama de bloques que ilustra un codificador de entropía de ejemplo 56 que puede implementar las técnicas descritas en esta divulgación. El codificador de entropía 56 recibe elementos de sintaxis, tales como uno o más elementos de sintaxis que representan la posición del último coeficiente de transformación significativo dentro de un bloque de coeficientes de transformación y codifica el elemento de sintaxis en un flujo de bits. Los elementos de sintaxis pueden incluir un elemento de sintaxis que especifica una coordenada-x de la posición de un último coeficiente significativo dentro de un bloque de coeficientes de transformación y un elemento de sintaxis que especifica una coordenada-y de la posición de un último coeficiente significativo dentro de un bloque de coeficientes de transformación. En un ejemplo, el codificador de entropía 56 ilustrado en la figura 5 puede ser un codificador CABAC. El codificador de entropía de ejemplo 56 en la figura 5 puede incluir una unidad de binarización 502, una unidad de codificación aritmética 504 y una unidad de asignación de contexto 506.

La unidad de binarización 502 recibe un elemento de sintaxis y produce una cadena bin. En un ejemplo, la unidad de binarización 502 recibe un valor que representa la última posición de un coeficiente significativo dentro de un bloque de coeficientes de transformación y produce una cadena de bits o un valor bin de acuerdo con los ejemplos descritos anteriormente. La unidad de codificación aritmética 504 recibe una cadena de bits procedente de la unidad de binarización 502 y realiza la codificación aritmética en la palabra de código. Según se muestra en la figura 5, el codificador aritmético 504 puede recibir valores bin procedentes de una ruta de derivación o la unidad de modelado de contexto 506. En el caso en que la unidad de codificación aritmética 504 recibe valores bin procedentes de la unidad de modelado de contexto 506, la unidad de codificación aritmética 504 puede realizar una codificación aritmética basada en asignaciones de contexto proporcionadas por la unidad de asignación de contexto 506. En un ejemplo, la unidad de codificación aritmética 504 puede usar asignaciones de contexto para codificar una parte de prefijo de una cadena de bits y puede codificar una parte de sufijo de una cadena de bits sin utilizar asignaciones de contexto.

- En un ejemplo, la unidad de asignación de contexto 506 puede asignar contextos en base a la indexación de contexto de ejemplo proporcionada en las Tablas 7-11 anteriores. De esta manera, el codificador de video 20 representa un codificador de video configurado para obtener un valor que indica una posición de un último coeficiente significativo dentro de un bloque de video de tamaño T, determinar una primera cadena de bits para el valor que indica la posición del último coeficiente significativo en base al esquema de codificación unaria truncada definido por una longitud máxima de bits definida por $2\log_2(T)-1$, $\log_2(T)+1$, o $\log_2(T)$, determinar una segunda cadena de bits para el valor que indica la posición del último coeficiente significativo en base a un esquema de codificación de longitud fija y codificar la primera y segunda cadenas de bits en un flujo de bits.
- 10 La figura 6 es un diagrama de flujo que ilustra un procedimiento de ejemplo para determinar una cadena de bits para un valor que indica la posición de un último coeficiente significativo de acuerdo con las técnicas de esta divulgación. El procedimiento descrito en la figura 6 puede ser realizado por cualquiera de los codificadores de video o codificadores de entropía de ejemplo descritos en este documento. En la etapa 602, se obtiene un valor que indica la posición de un último coeficiente de transformación significativo dentro de un bloque de video. En la etapa 604, se determina una
- 15 cadena de bits de prefijo para el valor que indica la posición del último coeficiente significativo. La cadena de bits de prefijo puede determinarse utilizando cualquiera de las técnicas descritas en este documento. En un ejemplo, el bits de prefijo se puede basar en un esquema de codificación unaria truncada definido por una longitud máxima de bits definida por $2\log_2(T)-1$, en el que T define el tamaño de un bloque de video. En otro ejemplo, el bits de prefijo se puede basar en un esquema de codificación unaria truncada definido por una longitud máxima de bits definida por $\log_2(T)+1$,
- 20 en el que T define el tamaño de un bloque de video. En otro ejemplo más, la cadena de bits de prefijo se puede basar en un esquema de codificación unaria truncada definido por una longitud máxima de bits definida por $\log_2(T)$, en el que T define el tamaño de un bloque de video. La cadena de bits de prefijo puede ser determinada por un codificador que realiza un conjunto de cálculos, por un codificador que utiliza tablas de búsqueda, o una combinación de los mismos. Por ejemplo, un codificador puede usar cualquiera de las Tablas 2, 5 y 6 para determinar la cadena de bits
- 25 de prefijo.
- En la etapa 606, se determina una cadena de bits de sufijo para el valor que indica la posición del último coeficiente significativo. La cadena de bits de sufijo puede determinarse utilizando cualquiera de las técnicas descritas en este documento. En un ejemplo, la cadena de bits de sufijo se puede basar en un esquema de codificación de longitud fija
- 30 definido por una longitud máxima de bits definida por $\log_2(T)-2$, en el que T define el tamaño de un bloque de video. En otro ejemplo, la cadena de bits de sufijo se puede basar en un esquema de codificación de longitud fija definido por una longitud máxima de bits definida por $\log_2(T)-1$, en el que T define el tamaño de un bloque de video. La cadena de bits de sufijo puede ser determinada por un codificador que realiza un conjunto de cálculos, por un codificador que utiliza tablas de búsqueda o una combinación de los mismos. Por ejemplo, un codificador puede usar cualquiera de
- 35 las Tablas 2, 5 y 6 para determinar la cadena de bits de sufijo. En la etapa 608, las cadenas de bits de prefijo y sufijo se codifican en un flujo de bits. En un ejemplo, las cadenas de bits de prefijo y sufijo pueden codificarse usando codificación aritmética. Cabe señalar que las partes de prefijo y de sufijo de un flujo de bits pueden intercambiarse. La codificación aritmética puede ser parte de un proceso de codificación CABAC o parte de otro proceso de codificación de entropía.
- 40
- Las Tablas 12-14 proporcionan un resumen de los resultados de simulación del rendimiento de codificación del esquema de binarización de ejemplo descrito con respecto a la Tabla 1 y del esquema de binarización de ejemplo descrito con respecto a la Tabla 2. Los resultados de simulación de las Tablas 12-14 fueron obtenidos utilizando condiciones de prueba comunes de alta eficiencia definidas por: F. Bossen, "Common test conditions and software
- 45 reference configurations", JCTVC-F900. Los valores negativos en las Tablas 12-14 indican una menor tasa de bits del esquema de binarización descrito con respecto a la Tabla 2 en comparación con el esquema de binarización descrito con respecto a la Tabla 1. Tiempo Cod. y Tiempo Decod. en las tablas 12-14 describen la cantidad de tiempo requerido para codificar y decodificar, respectivamente, el flujo de bits resultante del uso del esquema de binarización descrito con respecto a la Tabla 2 en comparación con la cantidad de tiempo requerido para codificar (o decodificar) el flujo de
- 50 bits resultante del uso del esquema de binarización descrito con respecto a la Tabla 1. Como se puede ver en los resultados experimentales mostrados en las Tablas 12-14, el esquema de binarización descrito con respecto a la Tabla 2 proporciona respectivas ganancias de rendimiento en la tasa BD de -0,04%, -0,01% y -0,03% en condiciones de prueba solo-intra de alta eficiencia, de acceso aleatorio y de bajo retardo.
- 55 Las clases A-E en las siguientes tablas representan varias secuencias de datos de video. Las columnas Y, U y V corresponden a datos de luma, croma-U y croma-V, respectivamente. La Tabla 12 resume estos datos para una configuración en la que todos los datos se codifican en modo intra. La Tabla 13 resume estos datos para una configuración en la que todos los datos se codifican en "acceso aleatorio" en la que están disponibles los modos intra e inter. La Tabla 14 resume estos datos para una configuración en la que las imágenes se codifican en un modo B de
- 60 bajo retardo.

TABLA 12-Todo Intra HE

	Y	U	V
Clase A	-0,03%	0,01%	0,02%
Clase B	-0,05%	-0,03%	-0,02%
Clase C	-0,02%	-0,01%	-0,02%
Clase D	-0,03%	-0,01%	-0,06%
Clase E	-0,08%	-0,04%	-0,01%
Todas	-0,04%	-0,02%	-0,02%
Tiempo Cod. [%]	99%		
Tiempo Decod. [%]	100%		

TABLA 13- Acceso aleatorio HE

	Y	U	V
Clase A	0,03%	0,05%	-0,07%
Clase B	-0,05%	0,04%	-0,21%
Clase C	-0,01%	-0,08%	-0,03%
Clase D	0,00%	0,01%	-0,12%
Clase E			
Todas	-0,01%	0,01%	-0,11%
Tiempo Cod. [%]	100%		
Tiempo Decod. [%]	100%		

5

TABLA 14-Delay bajo B HE

	Y	U	V
Clase A			
Clase B	-0,04%	-0,16%	-0,37%
Clase C	0,02%	-0,10%	-0,24%
Clase D	0,06%	-0,57%	-0,08%
Clase E	-0,21%	-0,65%	-0,66%
Todas	-0,03%	-0,05%	-0,07%
Tiempo Cod. [%]	99%		
Tiempo Decod. [%]	100%		

10

La figura 7 es un diagrama de bloques que ilustra un ejemplo de decodificador de video 30 que puede implementar las técnicas descritas en esta divulgación. En el ejemplo de la figura 7, el decodificador de video 30 incluye una unidad de decodificación de entropía 80, un módulo de predicción 81, una unidad de cuantificación inversa 86, un módulo de transformación inversa 88, un agregador 90 y una memoria de imágenes de referencia 92. El módulo de predicción 81 incluye una unidad de compensación de movimiento 82 y un módulo de intra-predicción 84. El decodificador de video 30 puede, en algunos ejemplos, realizar un paso de decodificación generalmente recíproco al paso de codificación descrito con respecto al codificador de video 20 de la figura 4.

15

Durante el proceso de decodificación, el decodificador de video 30 recibe un flujo de bits de video codificado que representa bloques de video de un segmento de video codificado y elementos de sintaxis asociados procedentes del codificador de video 20. La unidad de decodificación de entropía 80 del decodificador de video 30 decodifica por entropía el flujo de bits para generar coeficientes cuantificados, vectores de movimiento, y otros elementos de sintaxis. La unidad de decodificación de entropía 80 puede determinar un valor que indica la posición de un último coeficiente significativo dentro de un coeficiente de transformación en base a las técnicas descritas en este documento. La unidad de decodificación de entropía 80 reenvía los vectores de movimiento y otros elementos de sintaxis al módulo de predicción 81. El decodificador de video 30 puede recibir los elementos de sintaxis en el nivel de segmento de video y/o el nivel de bloque de video.

25

Cuando el segmento de video es codificado como un segmento intra-codificado (I), el módulo de intra-predicción 84 del módulo de predicción 81 puede generar datos de predicción para un bloque de video del segmento de video actual en base a un modo de intra-predicción señalizada y datos de bloques previamente decodificados de la trama o imagen actual. Cuando la trama de video es codificada como un segmento inter-codificado (es decir, B, P o GPB), la unidad de compensación de movimiento 82 del módulo de predicción 81 produce bloques predictivos para un bloque de video del segmento de video actual en base a los vectores de movimiento y otros elementos de sintaxis recibidos de la unidad de decodificación de entropía 80. Los bloques predictivos pueden producirse a partir de una de las imágenes

35

de referencia dentro de una de las listas de imágenes de referencia. El decodificador de video 30 puede construir las listas de tramas de referencia, Lista 0 y Lista 1, utilizando técnicas de construcción predeterminadas basadas en imágenes de referencia almacenadas en la memoria de imágenes de referencia 92.

5 La unidad de compensación de movimiento 82 determina información de predicción para un bloque de video del segmento de video actual analizando los vectores de movimiento y otros elementos de sintaxis, y usa la información de predicción para producir los bloques predictivos para el bloque de video actual que se está decodificando. Por ejemplo, la unidad de compensación de movimiento 82 utiliza algunos de los elementos de sintaxis recibidos para determinar un modo de predicción (por ejemplo, intra- o inter-predicción) utilizado para codificar los bloques de video del segmento de video, un tipo de segmento de inter-predicción (por ejemplo, segmento B, segmento P o segmento GPB), información de construcción para una o más de las listas de imágenes de referencia para el segmento, vectores de movimiento para cada bloque de video inter-codificado del segmento, estado de inter-predicción para cada bloque de video inter-codificado del segmento, y otra información para decodificar los bloques de video en el segmento de video actual.

15 La unidad de compensación de movimiento 82 también puede realizar una interpolación basada en filtros de interpolación. La unidad de compensación de movimiento 82 puede usar filtros de interpolación como los utilizados por el codificador de video 20 durante la codificación de los bloques de video para calcular valores interpolados para píxeles sub-enteros de bloques de referencia. En este caso, la unidad de compensación de movimiento 82 puede determinar los filtros de interpolación utilizados por el codificador de video 20 a partir de los elementos de sintaxis recibidos y utilizar los filtros de interpolación para producir bloques predictivos.

20 La unidad de cuantificación inversa 86 cuantifica de manera inversa, es decir, des-cuantifica, los coeficientes de transformación cuantificados proporcionados en el flujo de bits y decodificados por la unidad de decodificación de entropía 80. El proceso de cuantificación inversa puede incluir el uso de un parámetro de cuantificación calculado por el codificador de video 20 para cada bloque de video en el segmento de video para determinar un grado de cuantificación y, de la misma manera, un grado de cuantificación inversa que debe aplicarse. El módulo de transformación inversa 88 aplica una transformación inversa, por ejemplo, una DCT inversa, una transformación entera inversa, o un proceso de transformación inversa conceptualmente similar, a los coeficientes de transformación para producir bloques residuales en el dominio de píxeles.

Después de que el módulo de predicción 81 genere el bloque de predicción para el bloque de video actual en base a inter- o intra-predicción, el decodificador de video 30 forma un bloque de video decodificado agregando los bloques residuales generados por el módulo de transformación inversa 88 con los correspondientes bloques de predicción generados por el módulo de predicción 81. El agregador 90 representa el componente o componentes que realizan esta operación de agregación. Si se desea, también se puede aplicar un filtro de desbloqueo para filtrar los bloques decodificados con el fin de eliminar artefactos de bloqueo. También se pueden usar otros filtros de bucle (ya sea en el bucle de codificación o después del bucle de codificación) para suavizar las transiciones de píxeles, o mejorar la calidad del video. Los bloques de video decodificados en una determinada trama o imagen se almacenan luego en la memoria de imágenes de referencia 92, que almacena imágenes de referencia utilizadas para la compensación de movimiento subsiguiente. La memoria de imágenes de referencia 92 también almacena video decodificado para su presentación posterior en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la figura 1. De esta manera, el decodificador de video 30 representa un decodificador de video configurado para obtener de un flujo de bits codificado una primera cadena de bits y una segunda cadena de bits, determinar un valor que indica la posición de un último coeficiente significativo dentro de un bloque de video de tamaño T basándose en parte en la primera cadena de bits, en el que la primera cadena de bits es definida por un esquema de codificación unaria truncada con una longitud máxima de bits definida por $2\log_2(T)-1$ y determinar el valor que indica la posición del último coeficiente significativo basándose en parte en la segunda cadena de bits, en el que la segunda cadena de bits es definida por un esquema de codificación de longitud fija.

50 La figura 8 es un diagrama de flujo que ilustra un procedimiento de ejemplo para determinar un valor que indica la posición de un último coeficiente significativo dentro de un coeficiente de transformación a partir de una cadena de bits de acuerdo con las técnicas de esta divulgación. El procedimiento descrito en la figura 8 puede ser realizado por cualquiera de los decodificadores de video o unidades de decodificación de entropía de ejemplo descritos en este documento. En la etapa 802, se obtiene un flujo de bits codificado. Un flujo de bits codificado puede ser recuperado de una memoria o mediante una transmisión. El flujo de bits codificado puede ser codificado de acuerdo con un proceso de codificación CABAC u otro proceso de codificación de entropía. En la etapa 804, se obtiene una cadena de bits de prefijo. En la etapa 806, se obtiene una cadena de bits de sufijo. La cadena de bits de prefijo y la cadena de bits de sufijo se pueden obtener decodificando el flujo de bits codificado. La decodificación puede incluir la decodificación aritmética. La decodificación aritmética puede ser parte de un procesador de decodificación CABAC u otro proceso de decodificación por entropía. En la etapa 808, se determina un valor que indica una posición de un último coeficiente significativo dentro de un bloque de video de tamaño T. En un ejemplo, la posición de un último coeficiente significativo se determina basándose en parte en la cadena de bits de prefijo, en la que la cadena de bits de prefijo es definida por

- un esquema de codificación unaria truncada con una longitud máxima de bits definida por $2\log_2(T)-1$ en el que T define el tamaño de un bloque de video. En un ejemplo, la posición de un último coeficiente significativo se determina basándose en parte en la cadena de bits de prefijo, en el que la cadena de bits de prefijo es definida por un esquema de codificación unaria truncada con una longitud máxima de bits definida por $\log_2(T)+1$ en el que T define el tamaño de un bloque de video. En un ejemplo, la posición de un último coeficiente significativo se determina basándose en parte en la cadena de bits de prefijo, en el que la cadena de bits de prefijo es definida por un esquema de codificación unaria truncada con una longitud máxima de bits definida por $\log_2(T)$, en el que T define el tamaño de un bloque de video. En un ejemplo, la posición del último coeficiente significativo se determina basándose en parte en la cadena de bits de sufijo, en el que la cadena de bits de sufijo es definida por un esquema de codificación de longitud fija con una longitud máxima de bits definida por $\log_2(T)-2$, en el que T define el tamaño de un bloque de video. En otro ejemplo, la segunda cadena de bits se puede basar en un esquema de codificación de longitud fija definido por una longitud máxima de bits definida por $\log_2(T)-1$, en el que T define el tamaño de un bloque de video. Cabe señalar que las partes de prefijo y de sufijo de un flujo de bits pueden intercambiarse.
- 15 En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, firmware o cualquier combinación de los mismos. Si se implementa en software, las funciones pueden almacenarse en o transmitirse a través de, como una o más instrucciones o código, un medio legible informáticamente y ser ejecutadas por una unidad de procesamiento basada en hardware. Los medios legibles informáticamente pueden incluir medios de almacenamiento legibles informáticamente, 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 lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles informáticamente generalmente pueden corresponder a (1) medios de almacenamiento tangibles legibles informáticamente que no son transitorios o (2) un medio de comunicación tal como una señal u onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que pueden acceder uno o más sistemas informáticos 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 informáticamente.

- En aún otros ejemplos, la presente divulgación contempla un medio legible informáticamente que comprende una estructura de datos almacenada en el mismo, en el que la estructura de datos incluye un flujo de bits codificado compatible con esta divulgación. En particular, el flujo de bits codificado puede incluir un flujo de bits codificado por entropía que incluye una primera cadena de bits y una segunda cadena de bits, en el que la primera cadena de bits es indicativa de un valor que indica una posición del último coeficiente significativo y se basa en un esquema de codificación unaria truncada definido por una longitud máxima de bits definida por $2\log_2(T)-1$, y la segunda cadena de bits es indicativa de un valor que indica la posición del último coeficiente significativo y se basa en un esquema de codificación de longitud fija.

- A modo de ejemplo, y no como limitación, dichos medios de almacenamiento legibles informáticamente pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento en disco óptico, almacenamiento en disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que se pueda utilizar para almacenar el código de programa deseado en forma de instrucciones o estructuras de datos y que puedan ser accedidas por un sistema informático. Además, cualquier conexión se denomina correctamente un medio legible informáticamente. Por ejemplo, si las instrucciones se transmiten 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 como infrarrojos, radio y microondas se incluyen en la definición de medio. Sin embargo, debe entenderse que los medios de almacenamiento legibles informáticamente y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que están dirigidos a medios de almacenamiento tangibles, no transitorios. El disco, según se usa en este documento, incluye disco compacto (CD), disco láser, disco óptico, disco versátil digital (DVD), disquete y disco Blu-ray, en el que unos discos generalmente reproducen datos magnéticamente, mientras que otros discos reproducen datos ópticamente con láser. También deben incluirse combinaciones de los mismos dentro del alcance de los medios legibles informáticamente.

- Las instrucciones pueden ser ejecutadas por uno o más procesadores, tal como uno o más procesadores de señal digital (DSP), microprocesadores de propósito general, circuitos integrados específicos de aplicación (ASIC), matrices de lógica programable de campo (FPGA) u otro equivalente integrado o circuitos lógicos discretos. Por consiguiente, el término "procesador", según se usa en el presente documento, puede referirse a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en este documento se puede proporcionar dentro de hardware dedicado y/o módulos de software configurados para la codificación y decodificación, o incorporar 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, que incluyen un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de circuitos integrados (por ejemplo, un conjunto de chips). En esta divulgación se describen diversos componentes, módulos o unidades para enfatizar los aspectos funcionales de dispositivos configurados para realizar las técnicas descritas, pero no necesariamente requieren su realización por diferentes unidades de hardware. Más bien, según se ha descrito anteriormente, se pueden combinar varias unidades en una unidad de hardware de códec o ser proporcionadas por una colección de unidades de hardware interoperables, que incluyen uno o más procesadores según se ha descrito anteriormente, junto con software y/o firmware adecuados.

REIVINDICACIONES

1. Un procedimiento para codificar datos de video que comprende:
 5 obtener un valor que indica una posición de un último coeficiente significativo dentro de un bloque de video de tamaño T;
 determinar una binarización para el valor de la posición del último coeficiente significativo que consiste en una primera cadena de bits y una segunda cadena de bits;
 en el que la primera cadena de bits es determinada por un esquema de codificación unaria truncada;
 la segunda cadena de bits es determinada por un esquema de codificación de longitud fija; y
 10 codificar la primera y segunda cadenas de bits en un flujo de bits utilizando un proceso de codificación aritmética adaptativa al contexto (CABAC), comprendiendo el proceso CABAC codificar la primera cadena de bits en base a un modelo de contexto y codificar en modo bypass la segunda cadena de bits; **caracterizado porque** el esquema de codificación unaria truncada, mediante el cual se determina la primera cadena de bits, es definido por una longitud máxima de bits definida por $2\log_2(T)-1$.
- 15 2. El procedimiento de la reivindicación 1, en el que codificar en modo bypass la segunda cadena de bits comprende codificar bins de la segunda cadena de bits utilizando un estado de probabilidad igual.
3. Un procedimiento para decodificar datos de video que comprende:
 20 decodificar una primera cadena de bits y una segunda cadena de bits de un flujo de bits codificado utilizando un proceso de codificación aritmética adaptativa al contexto (CABAC), comprendiendo el proceso CABAC decodificar la primera cadena de bits en base a un modelo de contexto y decodificar en modo bypass la segunda cadena de bits;
 determinar un valor que indica la posición de un último coeficiente significativo dentro de un bloque de video de tamaño T en base a la primera cadena de bits y la segunda cadena de bits,
 25 en el que la primera cadena de bits es definida por un esquema de codificación unaria truncada con una longitud máxima de bits definida por $2\log_2(T)-1$; y
 en el que la segunda cadena de bits es definida por un esquema de codificación de longitud fija.
4. El procedimiento de la reivindicación 3, en el que decodificar en modo bypass la segunda cadena de bits comprende
 30 decodificar bins de la segunda cadena de bits utilizando un estado de probabilidad igual.
5. El procedimiento de una de las reivindicaciones 1 a 4, en el que el esquema de codificación de longitud fija es definido por una longitud máxima de bits definida por $\log_2(T)-2$.
- 35 6. El procedimiento de la reivindicación 5, en el que T es igual a 32, en el que el valor que indica la posición de un último coeficiente significativo es igual a 8, y en el que la primera cadena de bits tiene una longitud de bits igual a 7.
7. El procedimiento de una de las reivindicaciones 1 a 6, en el que el valor que indica la posición del último coeficiente significativo comprende uno de entre un valor que indica una posición-x del último coeficiente significativo o un valor
 40 que indica una posición-y del último coeficiente significativo.
8. Un dispositivo que comprende un codificador de video configurado para:
 obtener un valor que indica una posición de un último coeficiente significativo dentro de un bloque de video de tamaño T;
 45 determinar una binarización para el valor de la posición del último coeficiente significativo que consiste en una primera cadena de bits y una segunda cadena de bits;
 en el que la primera cadena de bits es determinada por un esquema de codificación unaria truncada;
 la segunda cadena de bits es determinada por un esquema de codificación de longitud fija; y
 codificar la primera y segunda cadenas de bits en un flujo de bits usando un proceso de codificación aritmética
 50 adaptativa al contexto (CABAC), comprendiendo el proceso CABAC codificar la primera cadena de bits en base a un modelo de contexto y codificar en modo bypass la segunda cadena de bits;
caracterizado porque el esquema de codificación unaria truncada, mediante el cual se determina la primera cadena de bits, es definido por una longitud máxima de bits definida por $2\log_2(T)-1$.
- 55 9. El dispositivo de la reivindicación 8, en el que para codificar en modo bypass la segunda cadena de bits, el codificador de video está configurado para codificar bins de la segunda cadena de bits utilizando un estado de probabilidad igual.
10. Un dispositivo que comprende un decodificador de video configurado para:
 60 decodificar una primera cadena de bits y una segunda cadena de bits de un flujo de bits codificado utilizando un proceso de codificación aritmética adaptativa al contexto (CABAC), comprendiendo el proceso CABAC decodificar la primera cadena de bits en base a un modelo de contexto y decodificar en modo bypass la segunda cadena de bits;

determinar un valor que indica la posición de un último coeficiente significativo dentro de un bloque de video de tamaño T en base a la primera cadena de bits y la segunda cadena de bits,

en el que la primera cadena de bits es definida por un esquema de codificación unaria troncada; y

en el que la segunda cadena de bits es definida por un esquema de codificación de longitud fija;

- 5 **caracterizado porque** el esquema de codificación unaria troncada, mediante el cual se determina la primera cadena de bits, es definido por una longitud máxima de bits definida por $2\log_2(T)-1$.

- 10 11. El dispositivo de la reivindicación 10, en el que para decodificar en modo bypass la segunda cadena de bits, el decodificador de video está configurado para decodificar bins de la segunda cadena de bits utilizando un estado de probabilidad igual.

12. El dispositivo de cualquiera de las reivindicaciones 8 a 11, en el que el esquema de codificación de longitud fija es definido por una longitud máxima de bits definida por $\log_2(T)-2$.

- 15 13. El dispositivo de la reivindicación 12, en el que T es igual a 32, en el que el valor que indica la posición de un último coeficiente significativo es igual a 8, y en el que la primera cadena de bits tiene una longitud de bits igual a 7.

- 20 14. El dispositivo de cualquiera de las reivindicaciones 8 a 13, en el que el valor que indica la posición del último coeficiente significativo comprende uno de entre un valor que indica una posición-x del último coeficiente significativo o un valor que indica una posición-y del último coeficiente significativo.

15. Un medio de almacenamiento legible informáticamente que comprende instrucciones almacenadas en el mismo que, cuando son ejecutadas, hacen que uno o más procesadores realicen el procedimiento de cualquiera de las reivindicaciones 1 a 7.

25

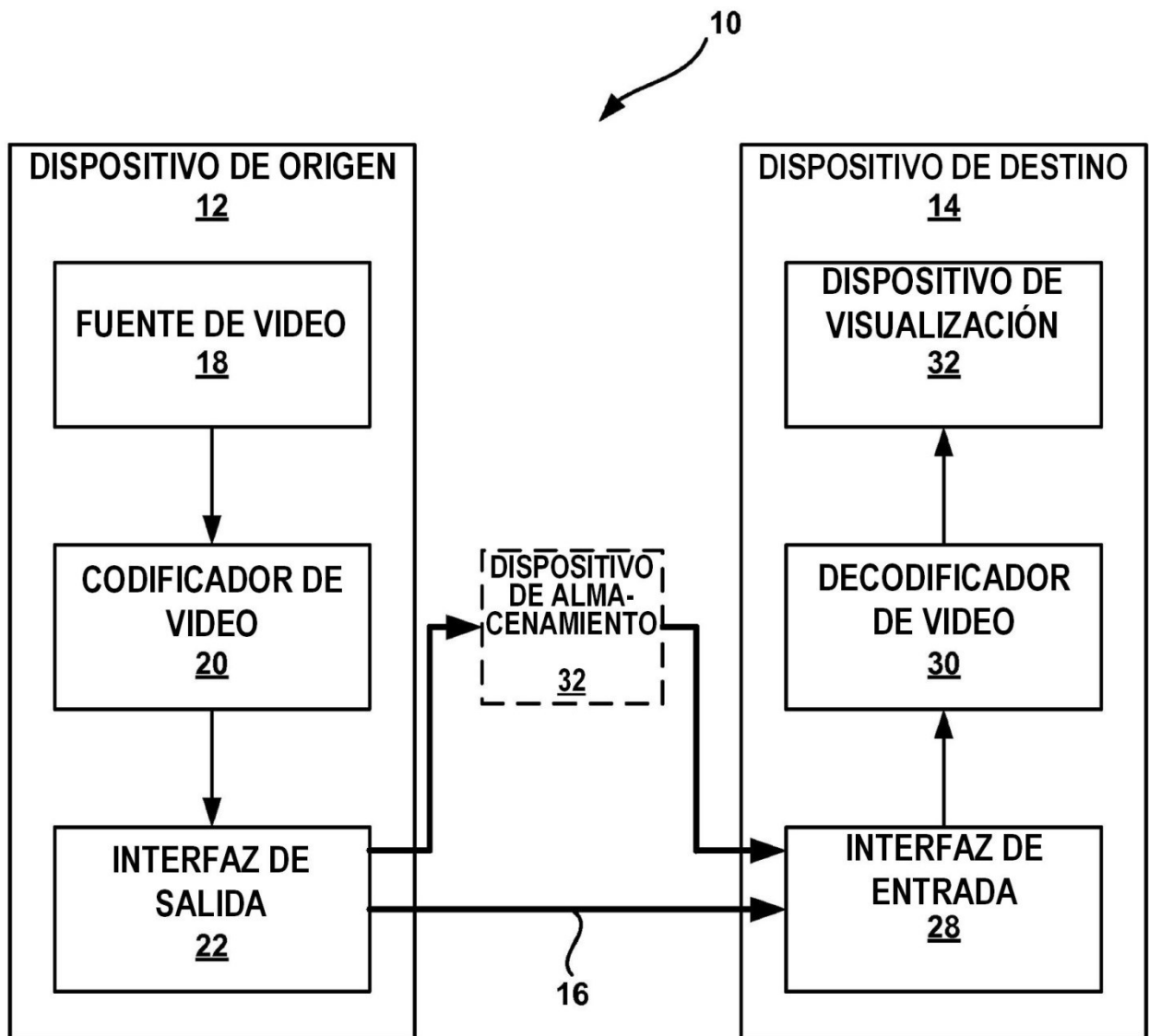


FIG. 1

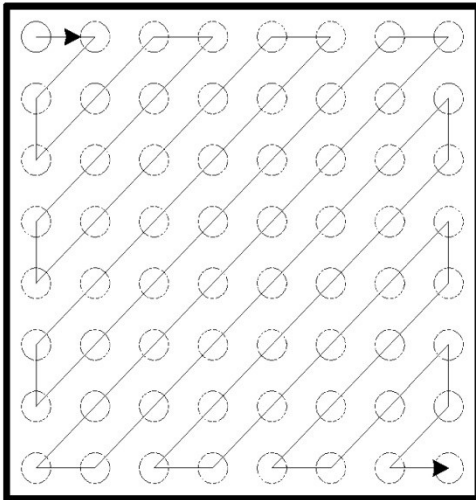


FIG. 2A

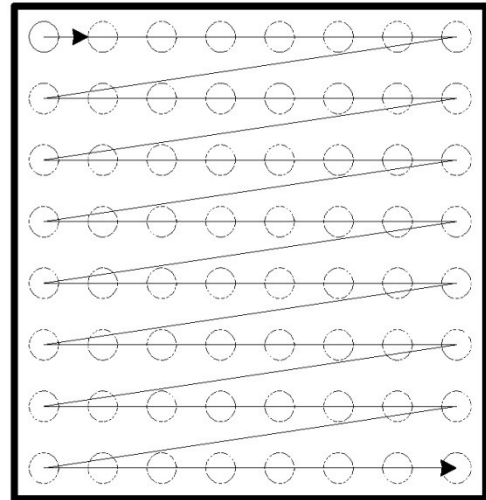


FIG. 2B

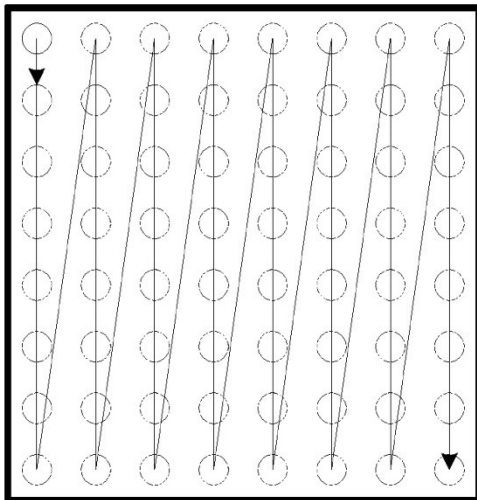


FIG. 2C

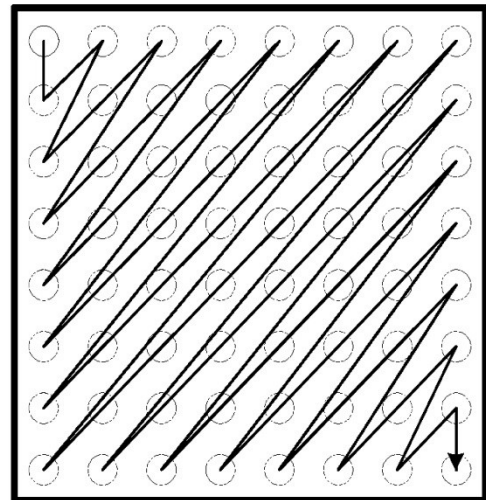


FIG. 2D

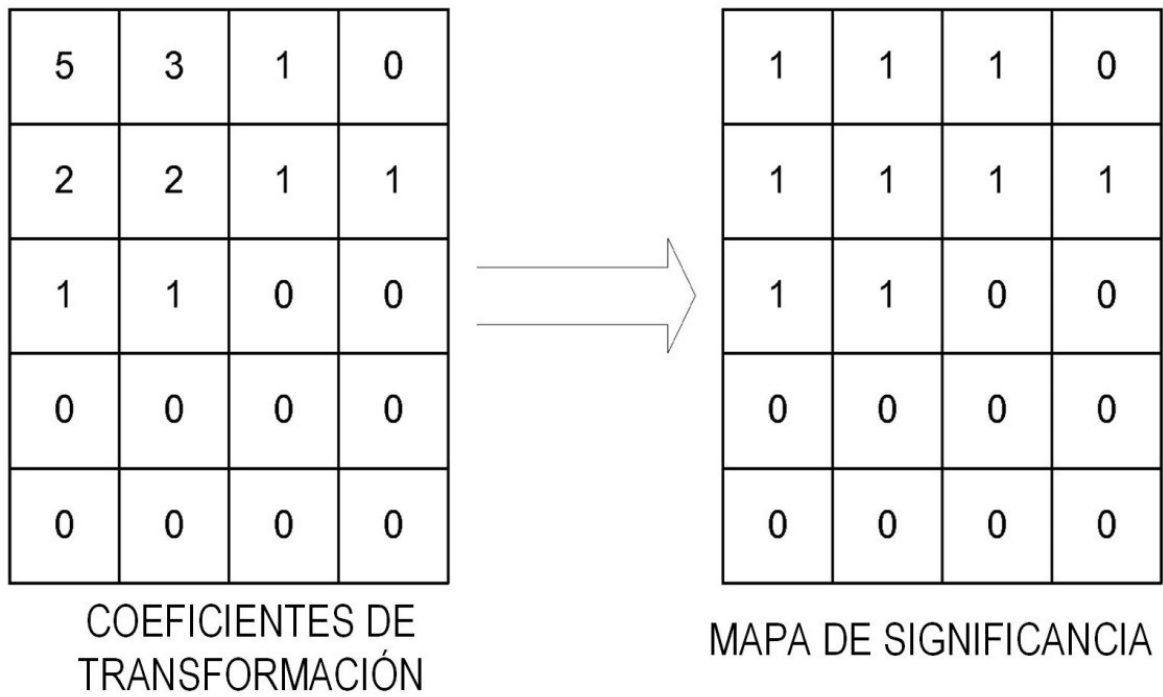


FIG. 3

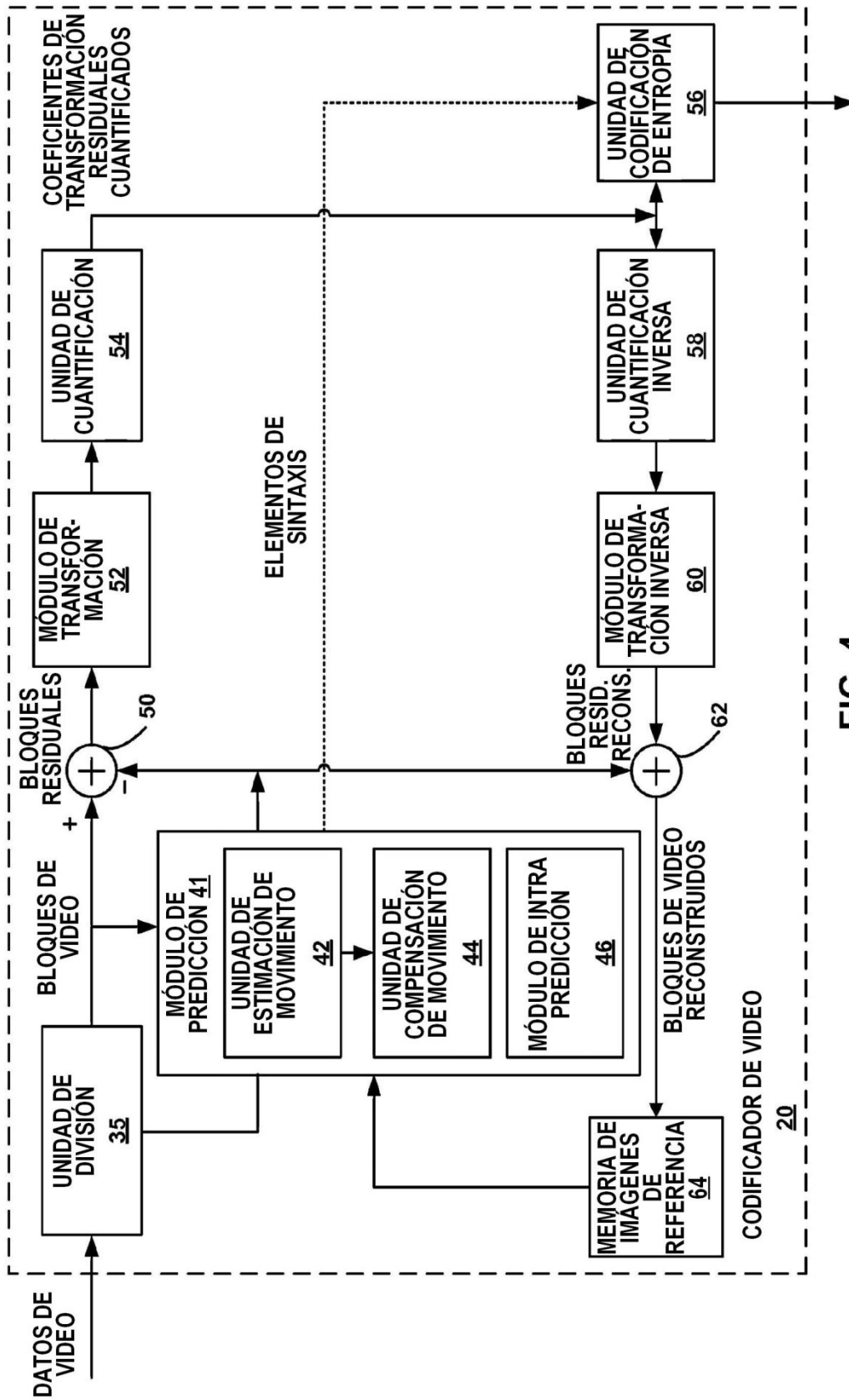


FIG. 4

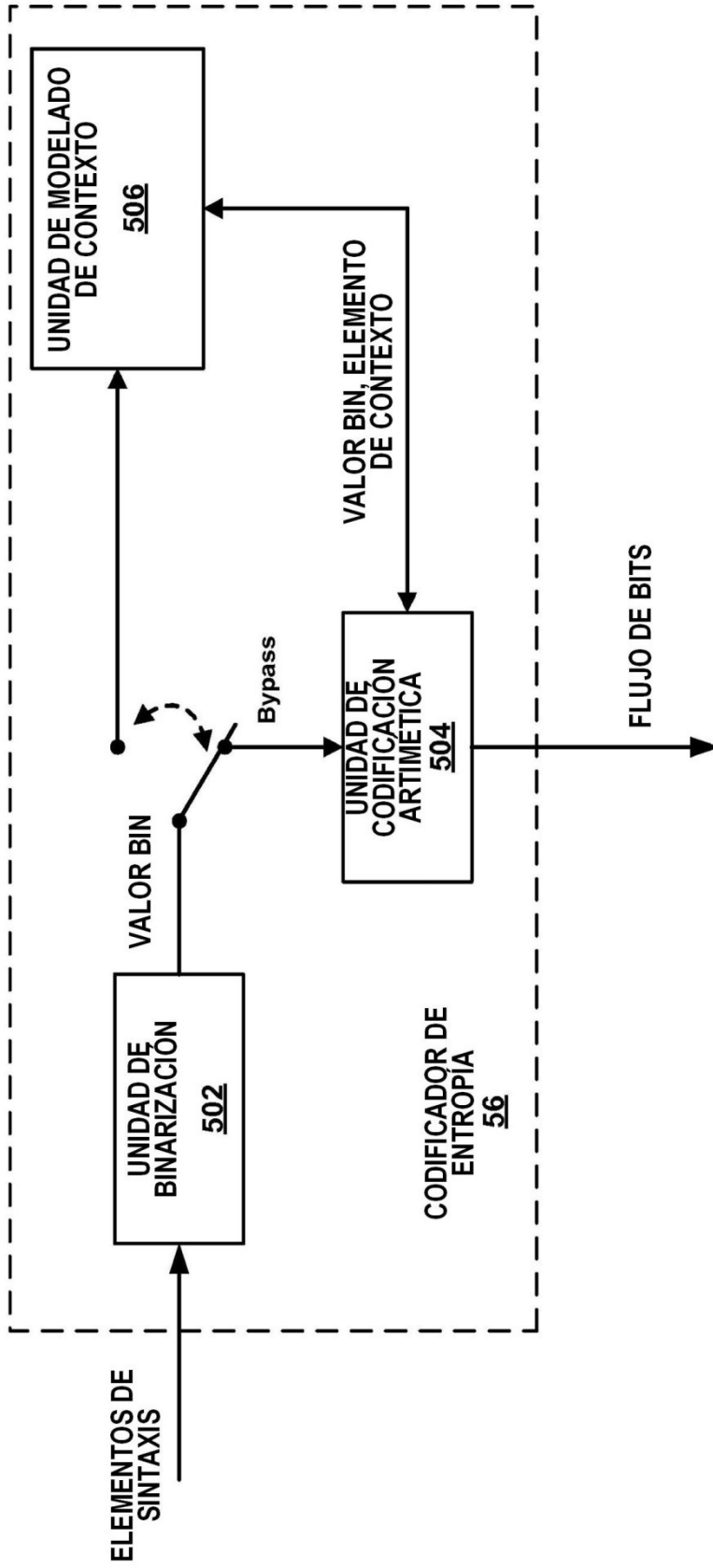


FIG. 5

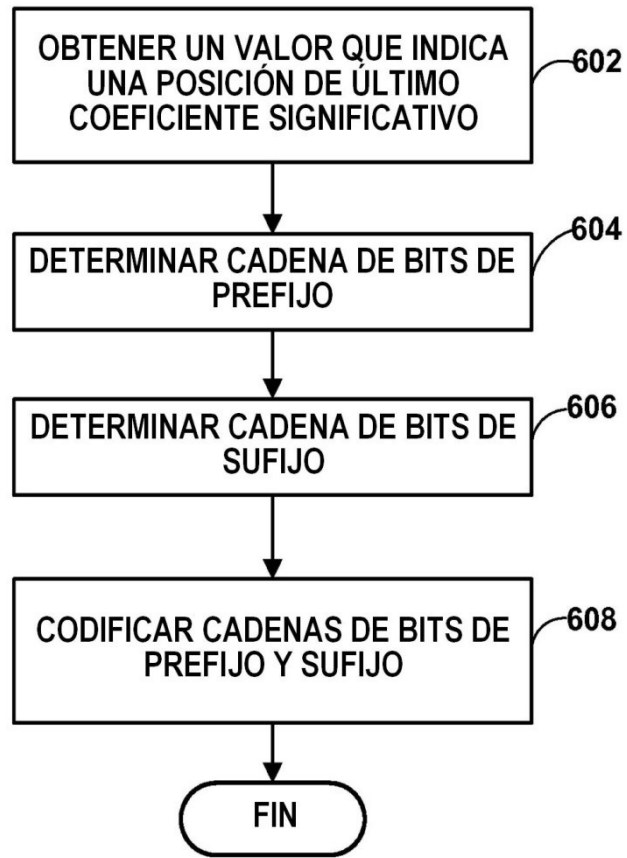


FIG. 6

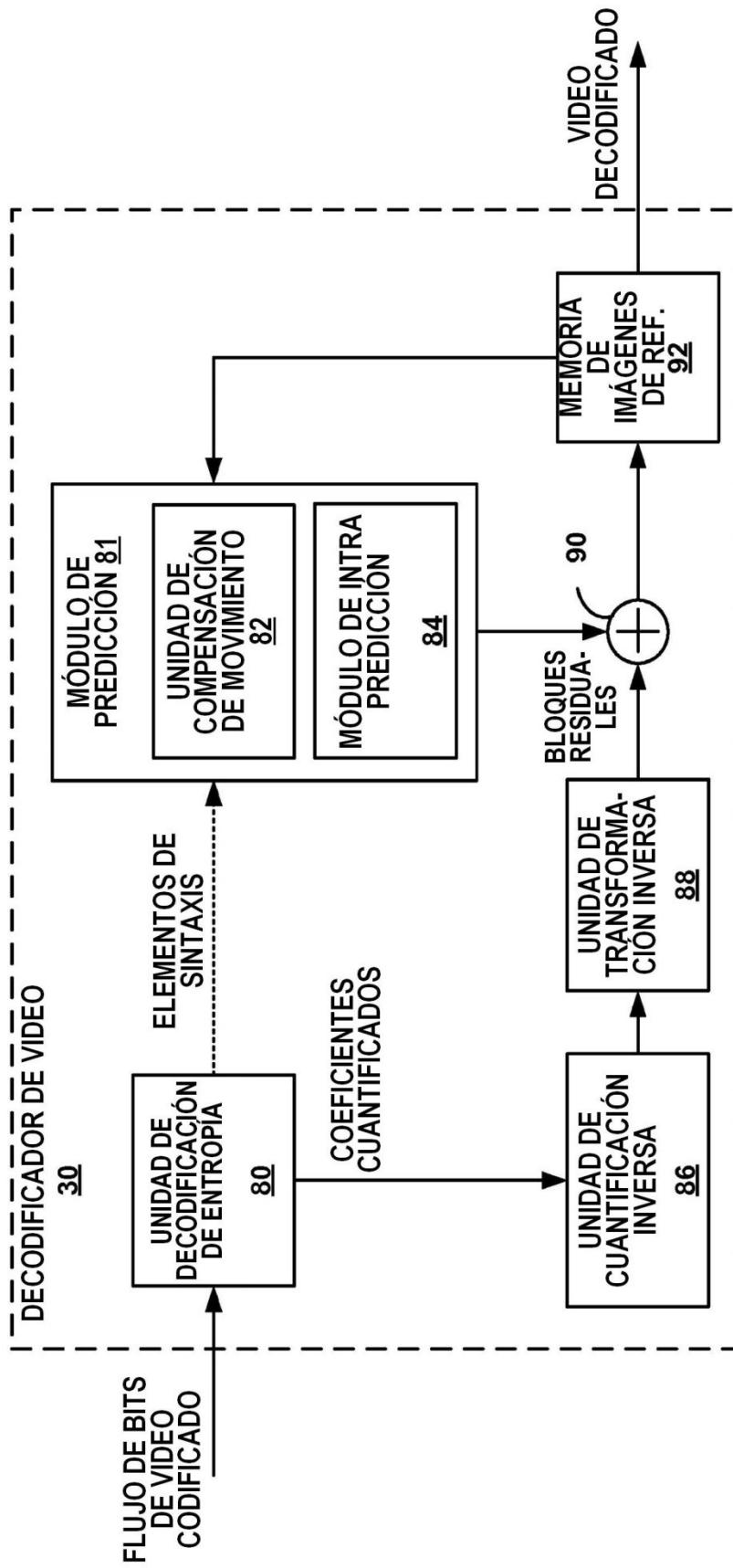


FIG. 7

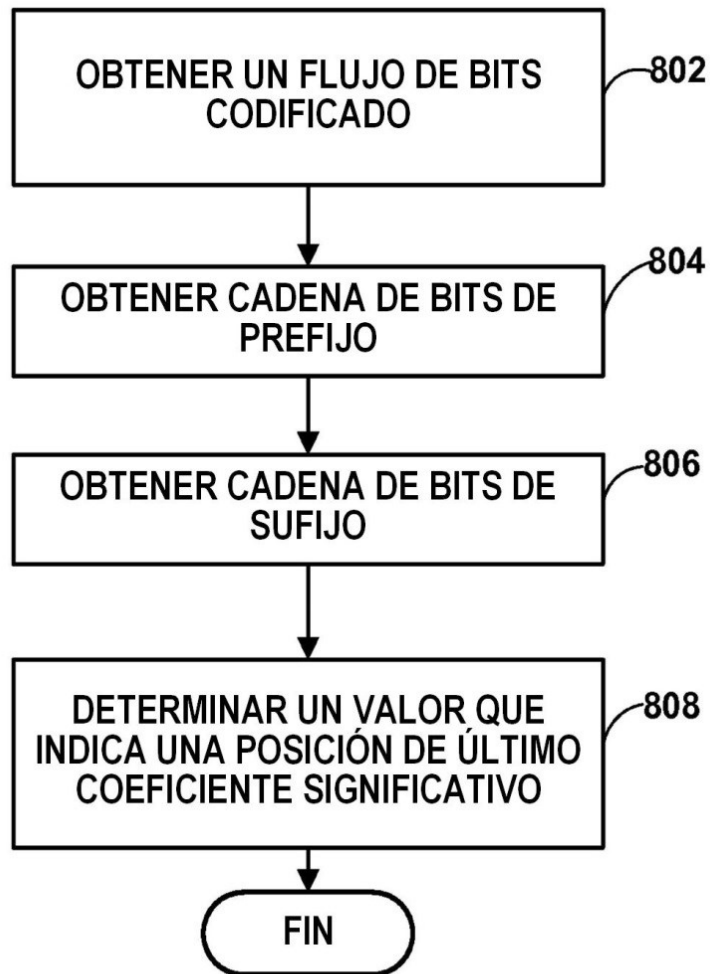


FIG. 8