

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 992 040**

51 Int. Cl.:

H03M 7/40 (2006.01)
H04N 19/13 (2014.01)
H04N 19/91 (2014.01)
H04N 19/103 (2014.01)
H04N 19/50 (2014.01)
H04N 19/174 (2014.01)
H04N 19/176 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **05.10.2012** **E 18196419 (8)**

97 Fecha y número de publicación de la concesión europea: **18.09.2024** **EP 3442126**

54 Título: **Número de reducción de contexto para la codificación aritmética binaria adaptativa al contexto**

30 Prioridad:

08.11.2011 US 201161557325 P
20.11.2011 US 201161561911 P
04.10.2012 US 201213645308

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
05.12.2024

73 Titular/es:

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

72 Inventor/es:

CHIEN, WEI-JUNG;
SOLE ROJAS, JOEL y
KARCZEWICZ, MARTA

74 Agente/Representante:

ISERN JARA, Jorge

ES 2 992 040 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

Número de reducción de contexto para la codificación aritmética binaria adaptativa al contexto

5 La presente solicitud reclama el beneficio de la Solicitud Provisional de los Estados Unidos núm. 61/557,325, presentada el 8 de noviembre de 2011, y la Solicitud Provisional de los Estados Unidos núm. 61/561,911, presentada el 20 de noviembre de 2011.

Campo técnico

10 Esta divulgación se refiere a la codificación de vídeo, y en particular a la codificación aritmética binaria adaptativa al contexto (CABAC) usada en la codificación de vídeo.

Antecedentes

15 Las capacidades de vídeo digital pueden incorporarse en una amplia gama de dispositivos, que incluye televisores digitales, sistemas de difusión digital directa, sistemas de difusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, tabletas, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, celular o
20 teléfonos por radio satélite, los denominados "teléfonos inteligentes", dispositivos de videoconferencia, dispositivos de transmisión de vídeo en directo y similares. 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 tales estándares. Los dispositivos de vídeo
25 pueden transmitir, recibir, codificar, decodificar y/o almacenar información de vídeo digital más eficientemente al implementar tales técnicas de compresión de vídeo.

Las técnicas de compresión de vídeo realizan predicciones espaciales (intraimagen) y/o predicciones temporales (interimagen) para reducir o eliminar la redundancia inherente a las secuencias de vídeo. Para la codificación de
30 vídeo basada en bloques, un segmento de vídeo (es decir, una imagen de vídeo o una porción de una imagen de vídeo) puede particionarse en bloques de vídeo, que también pueden denominarse bloques de árbol, unidades de codificación (CU) y/o nodos de codificación. Los bloques de vídeo en un segmento intracodificado (I) de una imagen se codifican mediante el uso de la predicción espacial con respecto a las muestras de referencia de los bloques vecinos de la misma imagen. Los bloques de vídeo en un segmento intercodificado (P o B) de una imagen pueden
35 usar predicción espacial con respecto a las muestras de referencia en bloques vecinos en la misma imagen o predicción temporal con respecto a las muestras de referencia en otras imágenes de referencia. Las imágenes pueden denominarse como cuadros, y las imágenes de referencia, pueden denominarse como cuadros de referencia.

La predicción espacial o temporal da como resultado un bloque predictivo para un bloque a codificar. Los datos residuales representan las diferencias de píxeles entre el bloque original a codificar y el bloque predictivo. Un bloque intercodificado se codifica de acuerdo con un vector de movimiento que apunta a un bloque de muestras de referencia que forma el bloque predictivo y los datos residuales indican la diferencia entre el bloque codificado y el
40 bloque predictivo. Un bloque intracodificado se codifica de acuerdo con un modo de intracodificación y los datos residuales. Para una compresión adicional, los datos residuales pueden transformarse del dominio de píxeles a un dominio de transformación, y dar como resultado coeficientes de transformación residuales, que luego pueden cuantificarse. Los coeficientes de transformación cuantificados, dispuestos inicialmente en una matriz bidimensional, pueden escanearse con el fin de producir un vector unidimensional de coeficientes de transformación, y puede aplicarse codificación de entropía para lograr una compresión aún mayor.

50 NPL 1
El Equipo de Colaboración Conjunta sobre la Codificación de Vídeo, JCTVC-G718, "Reducción de contexto para CABAC", describe una propuesta para reducir el número de contextos en CABAC.

55 NPL 2
El Equipo de Colaboración Conjunta sobre la Codificación de Vídeo, JCTVC-G785, "Codificación unificada de pred_type en CABAC", describe una propuesta para unificar la codificación de pred_type de los segmentos B y P en CABAC.

60 NPL 3
El Equipo de Colaboración Conjunta sobre la Codificación de Vídeo, JCTVC-G 151, "Binarización del modo de predicción y partición para P de bajo retardo", describe una corrección de errores y un procedimiento para binarizar los modos de predicción y partición para la configuración de P de bajo retardo.

65 NPL 4
El Equipo de Colaboración Conjunta sobre la Codificación de Vídeo, JCTVC-F803_d0, "WD4: Borrador de trabajo

4 de la codificación de vídeo de alta eficiencia", describe la representación codificada especificada en la sintaxis, diseñada para permitir una alta capacidad de compresión para una calidad de imagen o vídeo deseada.

Sumario

En general, esta divulgación describe técnicas para la codificación aritmética binaria adaptativa al contexto (CABAC) en un procedimiento de codificación de vídeo. En particular, esta divulgación propone una reducción en el número de contextos CABAC usados para uno o más elementos de sintaxis, cuyos ejemplos no limitantes incluyen *pred_type*, *merge_idx*, *inter_pred_flag*, *ref_idx_lx*, *cbf_cb*, *cbf_cr*, *coeff_abs_level_greater_flag*, y *coeff_abs_level_greater2_flag*. Las modificaciones pueden reducir hasta 56 contextos con cambios insignificantes en la eficiencia de codificación. Las reducciones de contexto propuestas para los elementos de sintaxis pueden usarse solas o en cualquier combinación.

En un ejemplo de la divulgación, un procedimiento de codificación de vídeo puede incluir determinar un primer tipo de predicción para un bloque de datos de vídeo en un segmento P, representar el primer tipo de predicción como un elemento de sintaxis de tipo de predicción de segmento P, determinar un segundo tipo de predicción para un bloque de datos de vídeo en un segmento B, representar el segundo tipo de predicción como un elemento de sintaxis de tipo de predicción de segmento B, determinar una binarización de segmento P para el elemento de sintaxis de tipo de predicción de segmento P, determinar una binarización de segmento B para el elemento de sintaxis de tipo de predicción de segmento B, en el que elemento de sintaxis de tipo de predicción de segmento P y el elemento de sintaxis de tipo de predicción de segmento B se determinan mediante el uso de la misma lógica de binarización, y codificar los datos de vídeo en base a las binarizaciones del elemento de sintaxis de tipo de predicción de segmento P y el elemento de sintaxis de tipo de predicción de segmento B.

En otro ejemplo de la divulgación, un procedimiento para decodificar vídeo puede incluir mapear un elemento de sintaxis de tipo de predicción de segmento P binarizado a un tipo de predicción mediante el uso de un mapeo de binarización para un bloque de datos de vídeo en un segmento P, mapear un elemento de sintaxis de tipo de predicción de segmento B binarizado a un tipo de predicción mediante el uso del mismo mapeo de binarización para un bloque de datos de vídeo en un segmento B, y decodificar los datos de vídeo en base a los tipos de predicción mapeados.

En otro ejemplo de la divulgación, un procedimiento de codificación de datos de vídeo comprende determinar un tipo de partición para un modo de predicción para un bloque de datos de vídeo, codificar un contenedor de tipo de partición de un elemento de sintaxis de tipo de predicción para un bloque de datos de vídeo mediante el uso de CABAC con un único contexto, en el que el único contexto es el mismo para cualquier tipo de partición, y codificar un contenedor de tamaño de partición del elemento de sintaxis de tipo de predicción para el bloque de datos de vídeo mediante el uso de CABAC en modo derivación.

En otro ejemplo de la divulgación, un procedimiento de decodificación de datos de vídeo comprende recibir un elemento de sintaxis de tipo de predicción para un bloque de datos de vídeo que se ha codificado mediante el uso de CABAC, incluyendo el elemento de sintaxis de tipo de predicción un contenedor de tipo de partición que representa un tipo de partición y un contenedor de tamaño de partición que representa un tamaño de partición, decodificar el contenedor de tipo de partición del elemento de sintaxis de tipo de predicción mediante el uso de la codificación aritmética binaria adaptativa al contexto con un único contexto, en el que el único contexto es el mismo para cualquier tipo de partición, y decodificar el contenedor de tamaño de partición del elemento de sintaxis de tipo de predicción mediante el uso de CABAC en modo derivación.

En otro ejemplo de la divulgación, un procedimiento de codificación de datos de vídeo comprende codificar un indicador de bloque codificado por croma Cb para un bloque de datos de vídeo mediante el uso de CABAC, en el que codificar el indicador de bloque codificado por croma Cb comprende usar un conjunto de contexto que incluye uno o más contextos como parte de CABAC, y codificar un indicador de bloque codificado por croma Cr mediante el uso de CABAC, en el que codificar el indicador de bloque codificado por croma Cr comprende usar el mismo conjunto de contexto que el indicador de bloque codificado por croma Cb como parte de la CABAC.

Esta divulgación también describe las técnicas anteriores en términos de aparatos configurados para realizar las técnicas, así como también en términos de un medio de almacenamiento legible por ordenador que almacena instrucciones que, cuando se ejecutan, hacen que uno o más procesadores realicen las técnicas.

Los detalles de uno o más ejemplos se establecen en las figuras adjuntas y la descripción más abajo. Otras características, objetivos y ventajas serán evidentes a partir de la descripción y las figuras y a partir de las reivindicaciones.

Breve descripción de las figuras

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

La Figura 2 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo que puede implementar las técnicas descritas en esta divulgación.

La Figura 3 es un diagrama de bloques que ilustra un ejemplo de decodificador de vídeo que puede implementar las técnicas descritas en esta divulgación.

La Figura 4 es un dibujo conceptual que muestra los tipos de particiones cuadradas y no cuadradas.

La Figura 5 es un dibujo conceptual que muestra los tipos de partición asimétrica.

La Figura 6 es un diagrama de flujo que ilustra un procedimiento de codificación de vídeo de ejemplo de la divulgación.

La Figura 7 es un diagrama de flujo que ilustra un procedimiento de decodificación de vídeo de ejemplo de la divulgación.

La Figura 8 es un diagrama de flujo que ilustra un procedimiento de codificación de vídeo de ejemplo de la divulgación.

La Figura 9 es un diagrama de flujo que ilustra un procedimiento de decodificación de vídeo de ejemplo de la divulgación.

La Figura 10 es un diagrama de flujo que ilustra un procedimiento de codificación de vídeo de ejemplo de la divulgación.

Descripción detallada

La presente divulgación describe técnicas para la codificación de datos, tal como los datos de vídeo. En particular, la divulgación describe técnicas que pueden promover una codificación eficiente de datos de vídeo mediante el uso de procedimientos de codificación por entropía adaptativa al contexto. Más específicamente, esta divulgación propone una reducción en el número de contextos CABAC usados para codificar elementos de sintaxis, tales como, *pred_type*, *merge_idx*, *inter_pred_flag*, *ref_idx_lx*, *cbf_cb*, *cbf_cr*, *coeff_abs_level(greater1_flag*, y *coeff_abs_level_greater2_flag*. Las modificaciones reducen hasta 56 contextos con cambios insignificantes en la eficiencia de codificación. Esta divulgación describe la codificación de vídeo con fines ilustrativos. Sin embargo, las técnicas descritas en esta divulgación también pueden aplicarse a la codificación de otros tipos de datos.

La Figura 1 es un diagrama de bloques que ilustra un sistema de codificación y decodificación de vídeo de ejemplo 10 que puede configurarse para utilizar técnicas para la codificación aritmética binaria adaptativa al contexto (CABAC) de acuerdo con ejemplos de la presente divulgación. Como se muestra en la Figura 1, el sistema 10 incluye un dispositivo de origen 12 que transmite el vídeo codificado al dispositivo de destino 14 a través del canal de comunicación 16. Los datos de vídeo codificados también pueden almacenarse en el medio de almacenamiento 34 o en el servidor de archivos 36 y puede accederse a ellos mediante el dispositivo de destino 14 como se desee. Cuando se almacena en un medio de almacenamiento o servidor de archivos, el codificador de vídeo 20 puede proporcionar datos de vídeo codificados a otro dispositivo, como una interfaz de red, un disco compacto (CD), Blu-ray o dispositivo de grabación o instalación de estampación de discos de vídeo digitales (DVD), u otros dispositivos, para almacenar los datos de vídeo codificados en el medio de almacenamiento. Igualmente, un dispositivo separado del decodificador de vídeo 30, como una interfaz de red, lector de CD o DVD, o similar, puede recuperar datos de vídeo codificados de un medio de almacenamiento y proporcionar los datos recuperados al decodificador de vídeo 30.

El dispositivo fuente 12 y el dispositivo de destino 14 pueden comprender cualquiera de una amplia variedad de dispositivos, que incluyen ordenadores de escritorio, notebook (es decir, ordenador portátil), tabletas, cajas decodificadoras, teléfonos móviles como los llamados teléfonos inteligentes, televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de vídeo juegos o similares. En muchos casos, estos dispositivos pueden equiparse para la comunicación inalámbrica. Por tanto, el canal de comunicación 16 puede comprender un canal inalámbrico, un canal alámbrico o una combinación de canales inalámbricos y alámbricos adecuados para la transmisión de datos de vídeo codificados. De manera similar, al servidor de archivos 36 puede accederse por el dispositivo de destino 14 a través de cualquier conexión de datos estándar, incluida una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión Wi-Fi), una conexión alámbrica (por ejemplo, DSL, módem por cable, etc.) o una combinación de ambos que sea adecuada para acceder a los datos de vídeo codificados almacenados en un servidor de archivos.

Las técnicas para CABAC, de acuerdo con ejemplos de esta divulgación, pueden aplicarse a la codificación de vídeo en apoyo de cualquiera de una variedad de aplicaciones multimedia, tales como difusión de la televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo en flujo, por ejemplo, a través de Internet, codificación de vídeo digital para su almacenamiento en un medio de almacenamiento de datos, decodificación de vídeo digital almacenado en un medio de almacenamiento de datos, u otras aplicaciones. En algunos ejemplos, el sistema 10 se puede configurar para soportar la transmisión de vídeo unidireccional o bidireccional para soportar aplicaciones tal como el vídeo en directo, la reproducción de vídeo, la difusión de vídeo, y/o la telefonía de vídeo.

En el ejemplo de la Figura 1, el dispositivo fuente 12 incluye la fuente de vídeo 18, el codificador de vídeo 20, el modulador/demodulador 22 y el transmisor 24. En el dispositivo fuente 12, la fuente de vídeo 18 puede incluir una fuente como un dispositivo de captura de vídeo, tal como una cámara de vídeo, un archivo de vídeo que contenga

vídeo capturado previamente, una interfaz de alimentación de vídeo para recibir vídeo de un proveedor de contenido de vídeo, y/o un sistema de gráficos por ordenador para generar datos de gráficos por ordenador como la fuente del vídeo, o una combinación de dichas fuentes. Como un ejemplo, si la fuente de vídeo 18 es una cámara de vídeo, el dispositivo fuente 12 y el dispositivo de destino 14 pueden formar los denominados teléfonos con cámara o videoteléfonos. Sin embargo, las técnicas descritas en esta divulgación pueden aplicarse a la codificación de vídeo en general, y pueden aplicarse a aplicaciones inalámbricas y/o alámbricas, o aplicaciones en las que los datos de vídeo codificados se almacenan en un disco local.

El vídeo capturado, precapturado o generado por ordenador puede ser codificado por el codificador de vídeo 20. La información de vídeo codificada puede modularse por el módem 22 de acuerdo con un estándar de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitirse al dispositivo de destino 14 a través del transmisor 24. El módem 22 puede incluir varios mezcladores, filtros, amplificadores u otros componentes diseñados para la modulación de la señal. El transmisor 24 puede incluir circuitos diseñados para transmitir datos, incluyendo amplificadores, filtros y una o más antenas.

El vídeo capturado, precapturado o generado por ordenador que se codifica por el codificador de vídeo 20 también puede almacenarse en el medio de almacenamiento 34 o en el servidor de archivos 36 para su consumo posterior. El medio de almacenamiento 34 puede incluir discos Blu-ray, DVD, CD-ROM, memoria flash o cualquier otro medio de almacenamiento digital adecuado para almacenar vídeo codificado. El vídeo codificado almacenado en el medio de almacenamiento 34 puede entonces accederse por el dispositivo de destino 14 para decodificación y reproducción. Aunque no se muestra en la Figura 1, en algunos ejemplos, el medio de almacenamiento 34 y/o el servidor de archivos 36 pueden almacenar la salida del transmisor 24.

El servidor de archivos 36 puede ser cualquier tipo de servidor capaz de almacenar vídeo codificado y transmitir ese vídeo codificado 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 en red (NAS), una unidad de disco local o cualquier otro tipo de dispositivo capaz de almacenar datos de vídeo codificados y transmitirlos a un dispositivo de destino. La transmisión de datos de vídeo codificados desde el servidor de archivos 36 puede ser una transmisión en tiempo real, una transmisión de descarga o una combinación de ambas. Al servidor de archivos 36 puede accederse por el dispositivo de destino 14 a través de cualquier conexión de datos estándar, incluida una conexión a internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión Wi-Fi), una conexión alámbrica (por ejemplo, DSL, módem por cable, Ethernet, USB, etc.), o una combinación de ambos que sea adecuada para acceder a datos de vídeo codificados almacenados en un servidor de archivos.

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

El dispositivo de visualización 32 puede integrarse con, o ser externo a, el dispositivo de destino 14. En algunos ejemplos, el dispositivo de destino 14 puede incluir un dispositivo de visualización integrado y también se configura 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 vídeo decodificados a un usuario, y puede comprender cualquiera de una variedad de dispositivos de visualización 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.

En el ejemplo de la Figura 1, el canal de comunicación 16 puede comprender cualquier medio de comunicación inalámbrico o cableado, como un espectro de radiofrecuencia (RF) o una o más líneas físicas de transmisión, o cualquier combinación de medios inalámbricos y cableados. El canal de comunicación 16 puede formar parte de una red basada en paquetes, como una red de área local, una red de área amplia o una red global como Internet. El canal de comunicación 16 generalmente representa cualquier medio de comunicación adecuado, o colección de diferentes medios de comunicación, para transmitir datos de vídeo desde el dispositivo fuente 12 al dispositivo de destino 14, que incluye cualquier combinación adecuada de medios alámbricos o inalámbricos. El canal de comunicación 16 puede incluir enrutadores, conmutadores, estaciones base, o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo fuente 12 al dispositivo de destino 14.

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar de acuerdo con un estándar de compresión de vídeo, como el estándar de codificación de vídeo de alta eficiencia (HEVC) actualmente en desarrollo por el Equipo de colaboración conjunta sobre codificación de vídeo (JCT-VC) del Grupo de Expertos en codificación de Vídeo (VCEG) de ITU-T y Grupo de Expertos en Imágenes en Movimiento ISO/IEC (MPEG). Un borrador reciente

de la norma HEVC, denominado "Borrador de trabajo HEVC 6" o "WD6", se describe en el documento JCTVC-H1003, Bross y otros, "Codificación de vídeo de alta eficiencia (HEVC) borrador 6 de la memoria descriptiva de texto," Equipo de Colaboración Conjunta sobre Codificación de vídeo (JCT-VC) de ITU-T SG16 WP3 y ISO/IEC JTC1/SC29/WG1 1, 8va reunión: San José, California, Estados Unidos, febrero de 2012, que, a partir del 1 de junio de 2012, se puede descargar desde http://phenix.int-evry.fr/jct/doc_end_user/documents/8_San%20Jose/wg11/JCTVC-H1003-v22.zip.

Alternativamente, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden operar de acuerdo con otros estándares propietarios o industriales, tal como el estándar UIT-T H.264, alternativamente denominado MPEG 4, Parte 10, Codificación Avanzada de Vídeo (AVC), o extensiones de tales estándares. Las técnicas de esta divulgación, sin embargo, no se limitan a ningún estándar de codificación particular. Otros ejemplos incluyen MPEG-2 e ITU-T H.263.

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

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementarse cada uno como cualquiera de una variedad de circuitos codificadores adecuados, como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados para aplicaciones específicas (ASIC), matrices de puertas lógicas programables en campo (FPGA), lógica discreta, software, hardware, microprograma o cualquiera de sus combinaciones. Cuando las técnicas se implementan de manera parcial en el software, un dispositivo puede almacenar instrucciones para el software en un medio legible por ordenador no transitorio adecuado y ejecutar las instrucciones en el hardware mediante el uso de uno o más procesadores para realizar las técnicas de esta divulgación. Cada uno de los codificadores de vídeo 20 y el decodificador de vídeo 30 puede incluirse en uno o más codificadores o decodificadores, cualquiera de los cuales puede integrarse como parte de un codificador/decodificador combinado (CODEC) en un dispositivo respectivo.

El codificador de vídeo 20 puede implementar cualquiera o todas las técnicas de esta divulgación para CABAC en un procedimiento de codificación de vídeo. Igualmente, el decodificador de vídeo 30 puede implementar cualquiera o todas estas técnicas para CABAC en un procedimiento de codificación de vídeo. Un codificador de vídeo, como se describió en esta divulgación, puede referirse a un codificador de vídeo o un decodificador de vídeo. De manera similar, una unidad de codificación de vídeo puede referirse a un codificador de vídeo o un decodificador de vídeo. Igualmente, la codificación de vídeo puede referirse a la codificación o decodificación de vídeo.

En un ejemplo de la divulgación, el codificador de vídeo 20 puede configurarse para determinar un primer tipo de predicción para un bloque de datos de vídeo en un segmento P, representar el primer tipo de predicción como un elemento de sintaxis de tipo de predicción de segmento P, determinar un segundo tipo de predicción para un bloque de datos de vídeo en un segmento B, representar el segundo tipo de predicción como un elemento de sintaxis de tipo de predicción de segmento B, determinar una binarización de segmento P para el elemento de sintaxis de tipo de predicción de segmento P, determinar una binarización de segmento B para el elemento de sintaxis de tipo de predicción de segmento B, en el que el elemento de sintaxis de tipo de predicción de segmento P y el elemento de sintaxis de tipo de predicción de segmento B se determinan mediante el uso de la misma lógica de binarización, y codificar los datos de vídeo en base a las binarizaciones del elemento de sintaxis de tipo de predicción de segmento P y el elemento de sintaxis de tipo de predicción de segmento B.

En otro ejemplo de la divulgación, el decodificador de vídeo 30 puede configurarse para mapear un elemento de sintaxis de tipo de predicción de segmento P binarizado a un tipo de predicción mediante el uso de un mapeo de binarización para un bloque de datos de vídeo en un segmento P, mapear un elemento de sintaxis de tipo de predicción de segmento B binarizado a un tipo de predicción mediante el uso del mismo mapeo de binarización para un bloque de datos de vídeo en un segmento B, y decodificar los datos de vídeo en base a los tipos de predicción mapeados.

En otro ejemplo de la divulgación, el codificador de vídeo 20 puede configurarse para determinar un tipo de partición para un modo de predicción para un bloque de datos de vídeo, codificar un contenedor de tipo de partición de un elemento de sintaxis de tipo de predicción para el bloque de datos de vídeo mediante el uso de CABAC con un único contexto, en el que el único contexto es el mismo para cualquier tipo de partición, y codificar un contenedor de tamaño de partición del elemento de sintaxis de tipo de predicción para el bloque de datos de vídeo mediante el uso de CABAC en modo derivación.

En otro ejemplo de la divulgación, el decodificador de vídeo 30 puede estar configurado para recibir un elemento de sintaxis de tipo de predicción para un bloque de datos de vídeo que ha sido codificado mediante el uso de CABAC, incluyendo el elemento de sintaxis de tipo de predicción un contenedor de tipo de partición que representa un tipo de

partición y un contenedor de tamaño de partición que representa un tamaño de partición, decodificando el contenedor de tipo de partición del elemento de sintaxis de tipo de predicción mediante el uso de CABAC con un único contexto, en el que el único contexto es el mismo para cualquier tipo de partición, y decodificando el contenedor de tamaño de partición del elemento de sintaxis de tipo de predicción mediante el uso de CABAC en modo derivación.

En otro ejemplo de la divulgación, tanto el codificador de vídeo 20 como el decodificador de vídeo 30 pueden configurarse para codificar un indicador de bloque codificado por croma Cb para un bloque de datos de vídeo mediante el uso de CABAC, en el que la codificación del indicador de bloque codificado por croma Cb comprende usar un conjunto de contexto que incluye uno o más contextos como parte de la CABAC, y codificar un indicador de bloque codificado por croma Cr mediante el uso de CABAC, en el que la codificación del indicador de bloque codificado por croma Cr comprende usar el mismo conjunto de contexto que el indicador de bloque codificado por croma Cb como parte de la CABAC.

El JCT-VC está trabajando en el desarrollo del estándar HEVC. Los esfuerzos de estandarización de HEVC son en base a un modelo en evolución de un dispositivo de codificación de vídeo denominado modelo de prueba HEVC (HM). El HM presupone diversas capacidades adicionales de los dispositivos de codificación de vídeo en relación con los dispositivos existentes de acuerdo con, por ejemplo, ITU-T H.264/AVC. Por ejemplo, mientras que H.264 proporciona nueve modos de codificación de intrapredicción, el HM puede proporcionar hasta treinta y tres modos de codificación de intrapredicción. En la siguiente sección se analizan con más detalles ciertos aspectos de la HM.

En general, el modelo de trabajo del HM describe que un cuadro o imagen de vídeo se puede dividir en una secuencia de bloques de árbol o unidades de codificación más grandes (LCU) que incluyen tanto las muestras de luma como de croma. Un bloque de árbol tiene un propósito similar al de un macrobloque del estándar H.264. Un segmento incluye una serie de bloques de árbol consecutivos en el orden de codificación. Un cuadro o una imagen de vídeo pueden particionarse en una 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 un nodo raíz del árbol cuádruple, se puede dividir en cuatro nodos hijo y cada nodo hijo puede ser a su vez un nodo padre y dividirse en otros cuatro nodos hijo. Un nodo hijo no dividido final, como un nodo hoja del árbol cuádruple, comprende un nodo de codificación, es decir, un bloque de vídeo codificado. Los datos de sintaxis asociados con un flujo de bits codificado pueden definir un número máximo de veces que un bloque de árbol se puede dividir y también pueden definir un tamaño mínimo de los nodos de codificación.

Una CU incluye un nodo de codificación y las unidades de predicción (PU) y las unidades de transformación (TU) asociadas con el nodo de codificación. Un tamaño de la CU generalmente corresponde al tamaño del nodo de codificación y, por lo general, debe tener forma cuadrada. El tamaño de la CU puede variar desde 8x8 píxeles hasta el tamaño del bloque de árbol con un máximo de 64x64 píxeles o superior. Cada CU puede contener una o más PU y una o más TU. Los datos de sintaxis asociados con una CU pueden describir, por ejemplo, la partición de la CU en una o más PU. Los modos de particionamiento pueden diferir entre si la CU es codificada en modo directo o en modo de salto, codificada en modo de intrapredicción o codificada en modo de interpredicción. Las PU pueden particionarse para que sean de forma no cuadrada. Los datos de sintaxis asociados con una CU también pueden describir, por ejemplo, la partición de la CU en una o más TU de acuerdo con un árbol cuaternario. Una TU puede ser de forma cuadrada o no cuadrada.

El emergente estándar HEVC permite transformaciones de acuerdo con las TU, que pueden ser diferentes para diferentes CU. El tamaño de las TU es usualmente en base al tamaño de las PU dentro de una CU determinada definida para una LCU particionada, aunque esto puede no ser siempre así. Las TU típicamente son del mismo tamaño o más pequeñas que las PU. En algunos ejemplos, las muestras residuales correspondientes a una CU se pueden subdividir en unidades más pequeñas mediante el uso de una estructura de árbol cuaternario conocida como "árbol cuaternario residual" (RQT). Los nodos hoja del RQT pueden denominarse como unidades de transformación (TU). Los valores de diferencia de píxeles asociados con las TU se pueden transformar para producir coeficientes de transformación, que pueden cuantificarse.

En general, una PU se refiere a datos relacionados con el procedimiento de predicción. Por ejemplo, cuando la PU se codifica dentro de modos, la PU puede incluir los datos que describen un modo de intrapredicción para la PU. Como otro ejemplo, cuando la PU se codifica entre modos, la PU puede incluir los datos que definen un vector de movimiento para la PU. Los datos que definen el vector de movimiento para una PU pueden describir, por ejemplo, un componente horizontal del vector de movimiento, un componente vertical del vector de movimiento, una resolución para el vector de movimiento (por ejemplo, precisión de un cuarto de píxel o precisión de un octavo de píxel), una imagen de referencia a la que apunta el vector de movimiento, y/o una lista de imágenes de referencia (por ejemplo, Lista 0, Lista 1 o Lista C) para el vector de movimiento.

En general, una TU se utiliza para los procedimientos de transformación y cuantificación. Una CU dada que tiene una o más PU también puede incluir una o más unidades de transformación (TU). Tras la predicción, el codificador de vídeo 20 puede calcular los valores residuales del bloque de vídeo identificado por el nodo de codificación de acuerdo con la PU. El nodo de codificación se actualiza entonces para hacer referencia a los valores residuales en

lugar del bloque de vídeo original. Los valores residuales comprenden valores de diferencia de píxeles que pueden transformarse en coeficientes de transformación, cuantificarse y escanearse mediante el uso de las transformaciones y otra información de transformación especificada en las TU para producir coeficientes de transformación serializados para la codificación de entropía. El nodo de codificación puede actualizarse de nuevo para hacer referencia a estos coeficientes de transformación serializados. Esta divulgación usa típicamente el término "bloque de vídeo" para referirse a un nodo de codificación de una CU. En algunos casos específicos, esta divulgación también puede utilizar el término "bloque de vídeo" para referirse a un bloque de árbol, es decir, LCU o CU, que incluye un nodo de codificación y las PU y las TU.

Una secuencia de vídeo típicamente incluye una serie de cuadros o imágenes de vídeo. Un grupo de imágenes (GOP) de manera general comprende una serie de una o más de las imágenes de vídeo. Un GOP puede incluir los datos de sintaxis en un encabezado del GOP, un encabezado de una o más de las imágenes o en cualquier otro lugar, que describa una serie de imágenes incluidas en el GOP. Cada segmento de una imagen puede incluir los datos de sintaxis del segmento que describen un modo de codificación para el segmento respectivo. El codificador de vídeo 20 típicamente opera en los bloques de vídeo dentro de los segmentos de vídeo individuales con el fin de codificar los datos de vídeo. Un bloque de vídeo puede corresponder a un nodo de codificación dentro de una CU. Los bloques de vídeo pueden tener tamaños fijos o variables y pueden diferir en tamaño de acuerdo con un estándar de codificación específico.

Como un ejemplo, el HM soporta la predicción en diversos tamaños de PU. Asumiendo que el tamaño de una CU particular es $2N \times 2N$, el HM soporta la intrapredicción en los tamaños de PU de $2N \times 2N$ o $N \times N$ y la interpredicción en los tamaños de PU simétricos de $2N \times 2N$, $2N \times N$, $N \times 2N$ o $N \times N$. El HM también soporta particiones asimétricas para la interpredicción en los tamaños de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$ y $nR \times 2N$. En la partición asimétrica, una dirección de una CU no se particiona, mientras que la otra dirección se particiona en 25 % y 75 %. La porción de la CU correspondiente a la partición del 25 % se indica mediante una "n" seguida de una indicación de "Arriba", "Abajo", "Izquierda" o "Derecha". Por tanto, por ejemplo, " $2N \times nU$ " se refiere a una CU de $2N \times 2N$ que se particiona de manera horizontal con una PU de $2N \times 0,5N$ en la parte superior y una PU de $2N \times 1,5N$ en la parte inferior.

La Figura 4 es un dibujo conceptual que muestra los tipos de partición cuadrados y no cuadrados para la intrapredicción y la interpredicción. La partición 102 es una partición $2N \times 2N$ y puede usarse tanto para la intrapredicción como para la interpredicción. La partición 104 es una partición $N \times N$ y puede usarse tanto para la intrapredicción como para la interpredicción. La partición 106 es una partición $2N \times N$ y actualmente se usa en HEVC para la interpredicción. La partición 108 es una partición $N \times 2N$ y actualmente se usa en HEVC para la interpredicción.

La Figura 5 es un dibujo conceptual que muestra los tipos de partición asimétrica. La partición 110 es una partición $2N \times nU$ y actualmente se usa en HEVC para la interpredicción. La partición 112 es una partición $2N \times nD$ y actualmente se usa en HEVC para la interpredicción. La partición 114 es una partición $nL \times 2N$ y actualmente se usa en HEVC para la interpredicción. La partición 116 es una partición $nR \times 2N$ y actualmente se usa en HEVC para la interpredicción.

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

Después de la codificación intrapredictiva o interpredictiva mediante el uso de las PU de una CU, el codificador de vídeo 20 puede calcular datos residuales a los que se aplican las transformadas especificadas por las TU de la CU. Los datos residuales pueden corresponder a las diferencias de píxeles entre los píxeles de la imagen no codificada y los valores de predicción correspondientes a las CU. El codificador de vídeo 20 puede formar los datos residuales para la CU y luego transformar los datos residuales para producir coeficientes de transformación.

Después de cualquier transformación para producir los coeficientes de transformación, el codificador de vídeo 20 puede realizar la cuantificación de los coeficientes de transformación. La cuantificación de manera general se refiere a un procedimiento en el que los coeficientes de transformación se cuantifican para de manera posible reducir la cantidad de datos utilizados para representar los coeficientes, proporcionando compresión adicional. El procedimiento de cuantificación puede reducir la profundidad de bits asociada con algunos o todos los coeficientes. Por ejemplo, un valor de bits n se puede redondear a un valor de bits m durante la cuantificación, donde n es mayor que m .

En algunos ejemplos, el codificador de vídeo 20 puede utilizar un orden de escaneo predefinido para escanear los coeficientes de transformación cuantificados para producir un vector serializado que puede codificarse por entropía.

En otros ejemplos, el codificador de vídeo 20 puede realizar un escaneo adaptativo. Después de escanear los coeficientes de transformación cuantificados para formar un vector unidimensional, el codificador de vídeo 20 puede codificar por entropía el vector unidimensional, por ejemplo, de acuerdo con la codificación de longitud variable adaptativa al contexto (CAVLC), la codificación aritmética binaria adaptativa al contexto (CABAC), la codificación aritmética binaria adaptativa al contexto basada en la sintaxis (SBAC), la codificación por Entropía de Particiones de Intervalos de Probabilidad (PIPE), o con otra metodología de codificación por entropía. El codificador de vídeo 20 también puede codificar por entropía los elementos de sintaxis asociados con los datos de vídeo codificados para su uso por el decodificador de vídeo 30 al decodificar los datos de vídeo.

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

Esta divulgación se relaciona con técnicas para codificadores de entropía de codificación aritmética binaria adaptativa al contexto (CABAC) u otros codificadores de entropía, como codificación de entropía de particiones de intervalo de probabilidad (PIPE) o codificadores relacionados. La codificación aritmética es una forma de codificación de entropía usada en muchos algoritmos de compresión que tienen una alta eficiencia de codificación, porque es capaz de mapear símbolos a palabras de código de longitud no entera. Un ejemplo de algoritmo de codificación aritmética es la codificación aritmética binaria basada en el contexto (CABAC) usada en H.264/AVC.

En general, la codificación de símbolos de datos mediante el uso de CABAC implica uno o más de las siguientes etapas:

- (1) Binarización: Si un símbolo a codificar tiene un valor no binario, es mapeado a una secuencia de los llamados "contenedores". Cada contenedor puede tener un valor de "0" o "1".
- (2) Asignación de contexto: Cada contenedor (en modo normal) se asigna a un contexto. Un modelo de contexto determina cómo se calcula un contexto para un contenedor dado en base a la información disponible para el contenedor, como los valores de los símbolos previamente codificados o el número de contenedor.
- (3) Codificación de contenedor: Los contenedores se codifican con un codificador aritmético. Para codificar un contenedor, el codificador aritmético requiere como entrada una probabilidad del valor del contenedor, es decir, una probabilidad de que el valor del contenedor sea igual a "0" y una probabilidad de que el valor del contenedor sea igual a "1". La probabilidad (estimada) de cada contexto se representa por un valor entero llamado "estado de contexto". Cada contexto tiene un estado y, por tanto, el estado (es decir, la probabilidad estimada) es el mismo para los contenedores asignados a un contexto y difiere entre los contextos.
- (4) Actualización de estado: La probabilidad (estado) de un contexto seleccionado se actualiza en base al valor codificado real del contenedor (por ejemplo, si el valor del contenedor era "1", aumenta la probabilidad de "1" 's').

Se debe señalar que la codificación de entropía de particiones de intervalo de probabilidad (PIPE) utiliza principios similares a los de la codificación aritmética y, por tanto, también puede utilizar las técnicas de esta divulgación.

CABAC en H.264/AVC y HEVC usa estados, y cada estado está implícitamente relacionado con una probabilidad. Existen variantes de CABAC, en las que se utiliza directamente la probabilidad de un símbolo ("0" o "1"), es decir, la probabilidad (o una versión entera de la misma) es el estado. Por ejemplo, tales variantes de CABAC se describen en "Descripción de la tecnología de codificación de vídeo propuesta por France Telecom, NTT, NTT DOCOMO, Panasonic y Technicolor", JCTVC-A114, 1ª reunión de JCT-VC, Dresden, DE, abril de 2010, denominado "JCTVC-A114" en lo sucesivo, y A. Alshin y E. Alshina, "Actualización de probabilidades multiparamétricas para CABAC," JCTVC-F254, 6ª reunión de JCT-VC, Turín, IT, julio de 2011, denominado "JCTVC-F254" en lo sucesivo.

En esta divulgación, se propone una reducción en el número de binarizaciones y/o contextos usados en CABAC. En particular, esta divulgación propone técnicas que pueden reducir el número de contextos usados en CABAC hasta en 56. Con 56 contextos menos, los resultados experimentales muestran cambios en la tasa de distorsión de bits (BD) de 0,00 %, 0,01 % y -0,13 % en condiciones de prueba de alta eficiencia, solo intra, acceso aleatorio y bajo retardo, respectivamente. Como tal, una reducción en la cantidad de contextos necesarios reduce las necesidades de almacenamiento tanto en el codificador como en el decodificador sin afectar materialmente la eficiencia de la codificación.

En esta divulgación, se reduce el número de contextos CABAC utilizados para los elementos de sintaxis, *pred_type*, *merge_idx*, *inter_pred_flag*, *ref_idx_lx*, *cbf_cb*, *cbl_cr*, *coeff_abs_level_greater1_flag*, and *coeff_abs_level_greater2_flag*, se propone. Las modificaciones reducen hasta 56 contextos con cambios insignificantes en la eficiencia de codificación. Las reducciones de contexto propuestas para los elementos de sintaxis anteriores pueden usarse solas o en cualquier combinación.

El elemento de sintaxis *pred_type* incluye un modo de predicción (*pred_mode_flag*) y un tipo de partición (*part_mode*) para cada unidad de codificación. El elemento de sintaxis *pred_mode_flag* igual a 0 especifica que la unidad de codificación actual se codifica en modo de interpredicción. El elemento de sintaxis *pred_mode_flag* igual a 1 especifica que la unidad de codificación actual se codifica en modo intrapredicción. El elemento de sintaxis *part_mode* especifica el modo de partición de la unidad de codificación actual.

El elemento de sintaxis *merge_idx*[x0][y0] especifica el índice de candidato de fusión de la lista de candidatos de fusión, donde x0, y0 especifican la ubicación (x0, y0) de la muestra de luma superior izquierda del bloque de predicción considerado en relación con la muestra de luma superior izquierda de la imagen. Cuando *merge_idx*[x0][y0] no está presente, se infiere que es igual a 0. Una lista de candidatos a fusión es una lista de unidades de codificación adyacentes a las unidades actuales desde las que puede copiarse información de movimiento.

El elemento de sintaxis *inter_pred_flag*[x0][y0] especifica si se usa predicción única o predicción doble para la unidad de predicción actual. Los índices de la matriz x0, y0 especifican la ubicación (x0, y0) de la muestra de luma superior izquierda del bloque de predicción considerado en relación con la muestra de luma superior izquierda de la imagen.

El elemento de sintaxis *ref_idx_lx* se refiere a una imagen de referencia específica dentro de una lista de imágenes de referencia.

Los elementos de sintaxis *cbf_cb*, *cbf_cr* indican si los bloques de transformación de croma (Cb y Cr, respectivamente) contienen coeficientes de transformación distintos de cero. El elemento de sintaxis *cbf_cb*[x0][y0][trafoDepth] igual a 1 especifica que el bloque de transformación Cb contiene uno o más niveles de coeficientes de transformación distintos de 0. Los índices de matriz x0, y0 especifican la ubicación (x0, y0) de la muestra de luma superior izquierda del bloque de transformación considerado en relación con la muestra de luma superior izquierda de la imagen. El índice de matriz *trafoDepth* especifica el nivel de subdivisión actual de una unidad de codificación en bloques con el propósito de realizar codificación de transformación. El índice de matriz *trafoDepth* es igual a 0 para los bloques que corresponden a unidades de codificación. Cuando *cbf_cb*[x0][y0][trafoDepth] no está presente y el modo de predicción no es intrapredicción, se infiere que el valor de *cbf_cb*[x0][y0][trafoDepth] es igual a 0.

El elemento de sintaxis *cbf_cr*[x0][y0][trafoDepth] igual a 1 especifica que el bloque de transformación Cr contiene uno o más niveles de coeficientes de transformación distintos de 0. Los índices de matriz x0, y0 especifican la ubicación (x0, y0) de la muestra de luma superior izquierda del bloque de transformación considerado en relación con la muestra de luma superior izquierda de la imagen. El índice de matriz *trafoDepth* especifica el nivel de subdivisión actual de una unidad de codificación en bloques con el propósito de realizar codificación de transformación. El índice de matriz *trafoDepth* es igual a 0 para los bloques que corresponden a unidades de codificación. Cuando *cbf_cr*[x0][y0][trafoDepth] no está presente y el modo de predicción no es intrapredicción, se infiere que el valor de *cbf_cr*[x0][y0][trafoDepth] es igual a 0.

El elemento de sintaxis *coeff_abs_level_greater1_flag*[n] especifica para la posición de escaneo n si hay niveles de coeficiente de transformación mayores que 1. Cuando *coeff_abs_level_greater1_flag*[n] no está presente, se infiere que es igual a 0.

El elemento de sintaxis *coeff_abs_level_greater2_flag*[n] especifica para la posición de escaneo n si hay niveles de coeficiente de transformación mayores que 2. Cuando *coeff_abs_level_greater2_flag*[n] no está presente, se infiere que es igual a 0.

En una propuesta para HEVC, diferentes binarizaciones en elementos de sintaxis *pred_type* se usan en los segmentos P y B como se muestra en la Tabla 1. Esta divulgación propone usar las mismas binarizaciones para los segmentos P y B. Se muestran ejemplos en las tablas 2-4. La Tabla 5 muestra el impacto del rendimiento de codificación en el segmento P en condiciones de prueba comunes (por ejemplo, ver F. Bossen, "Condiciones de prueba comunes y configuraciones de referencia de software", JCTVC-F900).

Tabla 1. Binarización para *pred_type* en una propuesta para HEVC

Tipo de segmento	Valor de <i>pred_type</i>	PredMode	PartMode	Cadena del contenedor		
				cLog2CUsSize > Log2MinCUsSize (1)	cLog2CUsSize = Log2MinCUsSize	
					cLog2CUsSize == 3 && !inter_4x4_enabled_flag (2)	cLog2CUsSize > 3 inter_4x4_enabled_flag (3)
I	0	MODE_INTRA	PART_2Nx2N	-	1	1
	1	MODE_INTRA	PART_NxN	-	0	0
P	0	MODE_INTER	PART_2Nx2N	0 1	0 1	0 1
	1	MODE_INTER	PART_2NxN	0 011	0 01	0 01
	2	MODE_INTER	PART_Nx2N	0 0011	0 00	0 001
	4	MODE_INTER	PART_2NxN _U	0 0100	-	-
	5	MODE_INTER	PART_2NxN _D	0 0101	-	-
	6	MODE_INTER	PART_nLx2N	0 00100	-	-
	7	MODE_INTER	PART_nRx2N	0 00101	-	-
	3	MODE_INTER	PART_NxN	-	-	0 000
	4	MODE_INTRA	PART_2Nx2N	1	11	11
	5	MODE_INTRA	PART_NxN	-	10	10
	0	MODE_INTER	PART_2Nx2N	1	1	1
	1	MODE_INTER	PART_2NxN	011	01	01
	2	MODE_INTER	PART_Nx2N	0011	001	001
B	4	MODE_INTER	PART_2NxN _U	0100	-	-
	5	MODE_INTER	PART_2NxN _D	0101	-	-
	6	MODE_INTER	PART_nLx2N	00100	-	-
	7	MODE_INTER	PART_nRx2N	00101	-	-
	3	MODE_INTER	PART_NxN	-	-	0001
	4	MODE_INTRA	PART_2Nx2N	000	000 0	0000 0
	5	MODE_INTRA	PART_NxN	-	000 1	0000 1

Como puede observarse en la Tabla 1, los segmentos I (por ejemplo, segmentos que solo incluyen bloques intrapredictivos) incluyen dos tipos de predicción diferentes (*pred_type*). Se usa una cadena del contenedor (binarización) para un bloque intrapredictivo con un tipo de partición 2Nx2N, y se usa otra cadena del contenedor para un bloque intrapredictivo con un tipo de partición NxN. Como se muestra en la Tabla 1, la cadena del contenedor usada para los segmentos I no depende del tamaño de CU.

Para los segmentos P y B, en la Tabla 1, se usan diferentes cadenas de contenedores para cada valor de *pred_type*. Nuevamente, el valor de *pred_type* depende tanto del modo de predicción (interpredicción o intrapredicción) como del tipo de partición usado. Para los segmentos P y B, la cadena del contenedor real usada depende además del tamaño de la CU que se está codificando y de si la interpredicción se habilita o no para un tamaño de bloque de 4x4.

La primera columna debajo de la cadena del contenedor se aplica a la situación donde la función logarítmica del

tamaño de CU de la CU que se está codificando es mayor que la función logarítmica del tamaño de CU mínimo permitido. De acuerdo con un ejemplo en HEVC, la primera columna de cadenas de contenedores se usa si $cLog2CUsSize > Log2MinCUsSize$. La función logarítmica se usa para crear un número más pequeño de modo que pueda usarse un índice consecutivo más pequeño.

Si la función logarítmica del tamaño de CU de la CU que se está codificando es equivalente a la función logarítmica del tamaño de CU mínimo permitido (es decir, $cLog2CUsSize = Log2MinCUsSize$), entonces se usa una de las columnas 2 y 3 bajo la cadena del contenedor en la Tabla 1 para seleccionar la binarización. La columna 2 se usa cuando la función logarítmica del tamaño de CU de la CU que se está codificando es equivalente a 3 y la intrapredicción mutua para una CU 4x4 no se habilita (es decir, $cLog2CUsSize = 3 \ \&\& \ !inter_4x4_enabled_flag$). La columna 3 se usa cuando la función logarítmica del tamaño de CU de la CU que se está codificando es mayor que 3 o cuando la interpredicción mutua para una CU 4x4 se habilita (es decir, $cLog2CUsSize > 3 \ || \ inter_4x4_enabled_flag$).

La Tabla 2 a continuación muestra ejemplos de binarizaciones donde los segmentos P y B usan las mismas cadenas de contenedores, de acuerdo con uno o más ejemplos descritos en esta divulgación. Como se muestra en la Tabla 2, los segmentos P usan las mismas binarizaciones usadas para los segmentos B en la Tabla 1. De esta manera, no es necesario almacenar y usar un conjunto separado de contextos para los segmentos P y B. Como tal, el número total de contextos necesarios para codificar el elemento de sintaxis *pred_type* se reduce. Además, solo es necesario almacenar un mapeo (en lugar de dos) entre la lógica de la cadena del contenedor (que se muestra en las columnas (1) - (3)) y la cadena del contenedor real.

Tabla 2. Binarización para *pred_type* en un ejemplo de la divulgación

Tipo de segmento	Valor de <i>pred_type</i>	PredMode	PartMode	Cadena del contenedor		
				$cLog2CUsSize > Log2MinCUsSize$ (1)	$cLog2CUsSize = Log2MinCUsSize$	
					$cLog2CUsSize = 3 \ \&\& \ !inter_4x4_enabled_flag$ (2)	$cLog2CUsSize > 3 \ \ inter_4x4_enabled_flag$ (3)
I	0	MODE_INTRA	PART_2Nx2N	-	1	1
	1	MODE_INTRA	PART_NxN	-	0	0
P o B	0	MODE_INTER	PART_2Nx2N	1	1	1
	1	MODE_INTER	PART_2NxN	011	01	01
	2	MODE_INTER	PART_Nx2N	0011	001	001
	4	MODE_INTER	PART_2NxN	0100	-	-
	5	MODE_INTER	PART_2NxND	0101	-	-
	6	MODE_INTER	PART_nLx2N	00100	-	-
	7	MODE_INTER	PART_nRx2N	00101	-	-
	3	MODE_INTER	PART_NxN	-	-	0001
	4	MODE_INTRA	PART_2Nx2N	000	000 0	0000 0
	5	MODE_INTRA	PART_NxN	-	000 1	0000 1

La Tabla 3 a continuación muestra otro ejemplo de binarización para *pred_type*. En este ejemplo, los segmentos B usan las mismas binarizaciones que los segmentos P de la Tabla 1. La Tabla 4 a continuación muestra un ejemplo adicional donde los segmentos P y segmentos B usan las mismas binarizaciones. Las tablas 2-4 solo tienen como objetivo mostrar ejemplos de binarizaciones compartidas entre los segmentos P y B. Puede usarse cualquier binarización o reglas de binarización para que los elementos de sintaxis *pred_type* de los segmentos P y B compartan las mismas binarizaciones.

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden almacenar las mismas reglas de mapeo y tablas de mapeo (por ejemplo, como se muestra en las Tablas 2-4) para su uso con los segmentos P y B. La codificación y decodificación CABAC puede aplicarse al elemento de sintaxis *pred_type* que usa estos mapeos.

De esta manera, el codificador de vídeo 20 puede configurarse para determinar un primer tipo de predicción para un bloque de datos de vídeo en un segmento P, representar el primer tipo de predicción como un elemento de sintaxis de tipo de predicción de segmento P, determinar un segundo tipo de predicción para un bloque de datos de vídeo en un segmento B, representar el segundo tipo de predicción como un elemento de sintaxis de tipo de predicción de segmento B, determinar una binarización de segmento P para el elemento de sintaxis de tipo de predicción de segmento P, determinar una binarización de segmento B para el elemento de sintaxis de tipo de predicción de segmento B, en el que el elemento de sintaxis de tipo de predicción de segmento P y el elemento de sintaxis de tipo de predicción de segmento B se determinan mediante el uso de la misma lógica de binarización, y codificar los datos de vídeo en base a las binarizaciones del elemento de sintaxis de tipo de predicción de segmento P y el elemento de sintaxis de predicción de segmento B.

El codificador de vídeo 20 puede configurarse además para binarizar el elemento de sintaxis de tipo de predicción de segmento P con la binarización de segmento P determinada, binarizar el elemento de sintaxis de tipo de predicción de segmento B con la binarización de segmento B determinada, aplicar codificación aritmética binaria adaptativa al contexto (CABAC) al elemento de sintaxis de tipo de predicción de segmento P binarizado y aplicar codificación aritmética binaria adaptativa al contexto (CABAC) al elemento de sintaxis de tipo de predicción de segmento B binarizado.

De manera similar, el decodificador de vídeo 30 puede configurarse para mapear un elemento de sintaxis de tipo de predicción del decodificador de vídeo P binarizado a un tipo de predicción mediante el uso de un mapeo de binarización para un bloque de datos de vídeo en un segmento P, mapear un elemento de sintaxis de tipo de predicción de segmento B binarizado a un tipo de predicción mediante el uso del mismo mapeo de binarización para un bloque de datos de vídeo en un segmento B, y decodificar los datos de vídeo en base a los tipos de predicción mapeados.

El decodificador de vídeo 30 puede configurarse además para recibir un elemento de sintaxis de tipo de predicción de segmento P codificado aritméticamente binario adaptativo al contexto que indica el tipo de predicción para el bloque de datos de vídeo en un segmento P, recibir un elemento de sintaxis de tipo de predicción de segmento B codificado aritméticamente binario adaptativo al contexto que indica el tipo de predicción para el bloque de datos de vídeo en un segmento B, decodificar el elemento de sintaxis de tipo de predicción de segmento P para producir el elemento de sintaxis de tipo de predicción de segmento P binarizado, y decodificar el elemento de sintaxis de tipo de predicción de segmento B para producir el elemento de sintaxis de tipo de predicción de segmento B binarizado.

Tabla 3. Binarización para *pred_type* en otro ejemplo de la divulgación

Tipo de segmento	Valor de <i>pred_type</i>	PredMode	PartMode	Cadena del contenedor		
				cLog2CUSize > Log2MinCUSize (1)	cLog2CUSize = Log2MinCUSize	
					cLog2CUSize == 3 && !inter_4x4_enabled flag (2)	cLog2CUSize > 3 inter_4x4_enabled flag (3)
I	0	MODE_INTRA	PART_2Nx2N	-	1	1
	1	MODE_INTRA	PART_NxN	-	0	0
P o B	0	MODE_INTER	PART_2Nx2N	0 1	0 1	0 1
	1	MODE_INTER	PART_2NxN	0 011	0 01	0 01
	2	MODE_INTER	PART_Nx2N	0 0011	0 00	0 001
	4	MODE_INTER	PART_2NxN	0 0100	-	-
	5	MODE_INTER	PART_2NxND	0 0101	-	-
	6	MODE_INTER	PART_nLx2N	0 00100	-	-
	7	MODE_INTER	PART_nRx2N	0 00101	-	-
	3	MODE_INTER	PART_NxN	-	-	0 000
	4	MODE_INTRA	PART_2Nx2N	1	11	11
	5	MODE_INTRA	PART_NxN	-	10	10

Tabla 4. Binarización para *pred_type* en otro ejemplo de la divulgación

Tipo de segmento	Valor de <i>pred_type</i>	PredMode	PartMode	Cadena del contenedor		
				cLog2CUSize > Log2MinCUSize (1)	cLog2CUSize = = Log2MinCUSize	
					cLog2CUSize = = 3 && !inter_4x4_enabled flag (2)	cLog2CUSize > 3 inter_4x4_enabled flag (3)
I	0	MODE_INTRA	PART_2Nx2N	-	1	1
	1	MODE_INTRA	PART_NxN	-	0	0
P o B	0	MODE_INTER	PART_2Nx2N	1	1	1
	1	MODE_INTER	PART_2NxN	011	01	01
	2	MODE_INTER	PART_Nx2N	001	00	001
	4	MODE_INTER	PART_2NxN	0100	-	-
	5	MODE_INTER	PART_2NxN	0101	-	-
	6	MODE_INTER	PART_nLx2N	0000	-	-
	7	MODE_INTER	PART_nRx2N	0001	-	-
	3	MODE_INTER	PART_NxN	-	-	000
	4	MODE_INTRA	PART_2Nx2N	000	000 0	0000 0
	5	MODE_INTRA	PART_NxN	-	000 1	0000 1

La Tabla 5 a continuación muestra el rendimiento de codificación mediante el uso de la binarización compartida para los segmentos P y B que se muestran en la Tabla 2. Como puede observarse en la Tabla 5, se pierde poca o ninguna eficiencia de codificación al usar las binarizaciones compartidas. P de bajo retardo HE (alta eficiencia) es una condición de prueba común para las binarizaciones del segmento (P) predicho unidireccionalmente. Las clases A-E representan diferentes resoluciones de cuadro. La clase A es una resolución de 2k x 4k. La clase B es una resolución de 1920 x 1080. La clase C es una resolución WVGA. La clase D es una resolución WQVGA. La clase E es una resolución de 720P. Un cambio del 0,1 al 0,2 por ciento en la condición de prueba P de bajo retardo HE generalmente se considera insignificante.

Tabla 5. Rendimiento de codificación para la binarización unificada en *pred_type*

Tasa de BD	P de bajo retardo HE		
	Y	U	V
Clase A			
Clase B	0,02 %	0,16 %	0,26 %
Clase C	0,01 %	0,05 %	-0,12 %
Clase D	-0,02 %	-0,10 %	-0,12 %
Clase E	0,02 %	0,03 %	0,05 %
Todo	0,01 %	0,04 %	0,03 %
Enc T[%]			
Dec T[%]			

De manera opcional, las mismas binarizaciones (no limitadas a las Tablas 2-4) para el tipo de predicción (incluye tamaño de predicción y/o modo de predicción) pueden compartirse en dos o más tipos diferentes de segmentos de predicción entre las predicciones. Los segmentos de predicción entre las predicciones pueden incluir, entre otras:

- Segmento P: Segmento que solo admite la predicción de movimiento unidireccional
- Segmento B: Segmento que admite la predicción de movimiento unidireccional y bidireccional
- En la codificación de vídeo escalable: la capa de mejora puede compartir las mismas binarizaciones con la capa base.
- En la codificación multivista: diferentes vistas pueden compartir las mismas binarizaciones.

Cuando se habilita la partición asimétrica, se usan cuatro contextos, divididos equitativamente en dos conjuntos de contexto, para CABAC en los dos últimos contenedores para señalar el elemento de sintaxis *pred_type* para las particiones asimétricas (es decir, PART_2NxN, PART_2NxND, PART_nLx2N, PART_nRx2N). En función de si la partición se divide a lo largo de la dirección horizontal o vertical, se aplica un conjunto de contexto. El penúltimo contenedor (es decir, el contenedor de tipo de partición; *part_mode*) especifica si la CU actual tiene particiones simétricas o particiones asimétricas. El último contenedor (es decir, el contenedor de tamaño de partición; *part_mode*) especifica si el tamaño de la primera partición es un cuarto o tres cuartos del tamaño de la CU. La Tabla 6 muestra un ejemplo de los contextos del penúltimo (tipo de partición) y del último (tamaño de partición) para el elemento de sintaxis *pred_type*.

Tabla 6. Contextos para los dos últimos contenedores del elemento de *sintaxisPred_Type*

Contenedor	Contexto
Tipo de partición (simétrica o asimétrica)	Conjunto de contexto 1 (2 contextos, uno para partición vertical, 1 para partición horizontal)
Tamaño de la partición (la primera partición es 1/4 CU o 3/4 CU)	Conjunto de contexto 2 (2 contextos, uno para 1/4 CU y otro para 3/4 CU)

Esta divulgación propone usar un contexto para el penúltimo contenedor (es decir, el contenedor de tipo de partición) y usar el modo derivación en el último contenedor (es decir, el contenedor de tamaño de partición). Como resultado, el número de contextos se reduce de 4 a 1. En la Tabla 7 se muestra un ejemplo del contexto usado de acuerdo con este ejemplo de la divulgación. La Tabla 8 muestra el rendimiento de codificación asociado con las modificaciones propuestas. El acceso aleatorio de alta eficiencia (HE) es una condición de prueba con cuadros de acceso aleatorio. B de bajo retardo HE es una condición de prueba que permite la predicción bidireccional.

Tabla 7. Contextos para los dos últimos contenedores del elemento de *sintaxisPred_Type* de acuerdo con un ejemplo de esta divulgación.

Contenedor	Contexto
Tipo de partición (simétrica o asimétrica)	Conjunto de contexto 1 (1 contexto)
Tamaño de la partición (la primera partición es 1/4 CU o 3/4 CU)	Modo derivación (sin contextos)

Tabla 8. Rendimiento de codificación del procedimiento propuesto para *pred_type*

Tasa de BD	Todo Intra HE			Acceso aleatorio HE			B de bajo retardo HE		
	Y	U	V	Y	U	V	Y	U	V
Clase A				0,03 %	-0,17 %	-0,29 %			
Clase B				0,02 %	-0,03 %	0,04 %	0,01 %	0,00 %	-0,24 %
Clase C				-0,01 %	-0,03 %	-0,02 %	-0,01 %	-0,03 %	0,02 %
Clase D				0,01 %	0,07 %	-0,05 %	0,01 %	0,06 %	0,03 %
Clase E							0,00 %	0,30 %	0,39 %
Todo				0,01 %	-0,04 %	-0,07 %	0,00 %	0,06 %	0,01 %
Enc T[%]									
Dec T[%]									

De esta manera, de acuerdo con este ejemplo, el codificador de vídeo 20 puede configurarse para determinar un tipo de partición para un modo de predicción para un bloque de datos de vídeo, codificar un contenedor de tipo de partición de un elemento de sintaxis de tipo de predicción para el bloque de datos de vídeo mediante el uso de la codificación aritmética binaria adaptativa al contexto con un único contexto, en el que el único contexto es el mismo para cualquier tipo de partición, y codificar un contenedor de tamaño de partición de una sintaxis de tipo de predicción para el bloque de datos de vídeo mediante el uso de la codificación aritmética binaria adaptativa al contexto en modo derivación.

De manera similar, de acuerdo con este ejemplo, el decodificador de vídeo 30 puede configurarse para recibir un elemento de sintaxis de tipo de predicción para un bloque de datos de vídeo que se ha codificado mediante el uso de la codificación aritmética binaria adaptativa al contexto (CABAC), incluyendo el elemento de sintaxis de tipo de predicción un contenedor de tipo de partición que representa un tipo de partición y un contenedor de tamaño de partición que representa un tamaño de partición, decodificando el contenedor de tipo de partición del elemento de sintaxis de tipo de predicción mediante el uso de la codificación aritmética binaria adaptativa al contexto con un único contexto, en el que el único contexto es el mismo para cualquier tipo de partición, y decodificando el contenedor de tamaño de partición de la sintaxis de tipo de predicción para un bloque de datos de vídeo mediante el uso de la codificación aritmética binaria adaptativa al contexto en modo derivación.

En otro ejemplo, al codificar un tipo de partición rectangular, puede usarse el modo derivación o un único contexto para el contenedor que indica si el modo de partición es PART_nLx2N o PART_nRx2N, o si el modo es

PART_2NxN_U, PART_2NxN_D. El uso del modo derivación o de un único contexto es aplicable porque la probabilidad de que se use cualquiera de los modos de partición es cercana al 50 %. También de manera opcional, puede usarse el modo derivación o un único contexto para el contenedor que indica si el modo es una partición simétrica o una partición asimétrica.

El siguiente ejemplo de la divulgación se relaciona con la señalización en un modo de "fusión" de interpredicción. En el modo de fusión, el codificador ordena a un decodificador, a través de la señalización del flujo de bits de la sintaxis de predicción, que copie un vector de movimiento, un índice de referencia (que identifica una imagen de referencia, en una lista de imágenes de referencia dada, a la que apunta el vector de movimiento) y la dirección de predicción de movimiento (que identifica la lista de imágenes de referencia (Lista 0 o Lista 1), es decir, en términos de si la imagen de referencia precede o sigue temporalmente al cuadro actual) de un vector de movimiento candidato seleccionado para una porción actual de la imagen que va a codificarse. Esto se consigue al señalar en el flujo de bits un índice en una lista de vectores de movimiento candidatos que identifica el vector de movimiento candidato seleccionado (es decir, el candidato a predictor del vector de movimiento espacial (MVP) o el candidato a MVP temporal).

Por tanto, para el modo fusión, la sintaxis de predicción puede incluir un indicador que identifique el modo (en este caso el modo "fusión") y un índice (*merge_idx*) que identifique el vector de movimiento candidato seleccionado. En algunos casos, el vector de movimiento candidato estará en una porción causal en referencia a la porción actual. Es decir, el decodificador ya decodificó el vector de movimiento candidato. Como tal, el decodificador ya recibió y/o determinó el vector de movimiento, el índice de referencia y la dirección de predicción de movimiento para la porción causal. Como tal, el decodificador puede simplemente recuperar el vector de movimiento, el índice de referencia y la dirección de predicción de movimiento asociados con la porción causal de la memoria y copiar estos valores como información de movimiento para la porción actual. Para reconstruir un bloque en modo de fusión, el decodificador obtiene el bloque predictivo mediante el uso de la información de movimiento derivada para la porción actual y agrega los datos residuales al bloque predictivo para reconstruir el bloque codificado.

En HM4.0, uno de los cinco candidatos a fusión se señala cuando la PU actual está en modo fusión. Se utiliza un código unario truncado para representar el elemento de sintaxis *merge_idx*. En una propuesta para HEVC, para CABAC, cada contenedor usa un contexto. Esta divulgación propone usar un contexto repetidamente en los cuatro contenedores, como se muestra en la Tabla 9.

Tabla 9. Contextos para los dos últimos contenedores del elemento de *syntaxisPred_Type*

Contenedor	Contexto
Contenedor 0-3 para <i>merge_idx</i>	Conjunto de contexto 1 (el mismo conjunto de contexto para todos los contenedores)

La Tabla 10 muestra el rendimiento de codificación asociado con este ejemplo.

Tabla 10. Rendimiento de codificación del procedimiento propuesto en *merge_idx*

Tasa de BD	Todo Intra HE			Acceso aleatorio HE			B de bajo retardo HE		
	Y	U	V	Y	U	V	Y	U	V
Clase A				0,00 %	-0,20 %	-0,07 %			
Clase B				0,01 %	0,03 %	0,03 %	0,01 %	-0,08 %	-0,22 %
Clase C				0,00 %	0,04 %	0,00 %	0,00 %	-0,08 %	-0,09 %
Clase D				0,03 %	-0,09 %	-0,05 %	0,05 %	-0,24 %	0,44 %
Clase E							-0,14 %	0,08 %	0,78 %
Todo				0,01 %	-0,05 %	-0,02 %	-0,01 %	-0,09 %	0,17 %
Enc T[%]									
Dec T[%]									

Opcionalmente, puede usarse más de un contexto en la codificación del índice de fusión, con algunos contenedores compartiendo el mismo contexto y otros contenedores usando otros contextos. A modo de ejemplo, sólo los contenedores consecutivos comparten el mismo contexto. Por ejemplo, contenedor2 y contenedor3 pueden compartir un contexto; contenedor2 y contenedor4 no pueden compartir el mismo contexto a menos que el contenedor3 también comparta el contexto.

Como otro ejemplo, supongamos que el número total de contenedores del índice de fusión es N (el primer contenedor es contenedor0, el último contenedor es contenedor N-1). Los umbrales Y, *thres_i*, *i*=1,...,y, se usan para determinar el contexto compartido en la codificación del índice de fusión. En este ejemplo, las siguientes reglas indican cómo se comparten los contextos entre contenedores:

1. $0 < Y < N$ (hay menos umbrales que contenedores)
2. $thres_i < thres_{i+1}$

3. $0 < thres_i$
4. $thres_Y = N$
5. $contenedor_j$ compartirá un contexto donde $i = \{thres_Y, \dots, thres_{i+1}-1\}$

5 En base a estas reglas, el procedimiento anterior en el que un contexto se usa repetidamente en los cuatro contenedores, puede verse como un caso donde $N=4$, $Y=1$, $thres_i=4$. Por lo tanto, del contenedor 0 al contenedor 3 comparten el mismo contexto.

10 Otro ejemplo incluye establecer $N=4$, $Y=2$, $thres_i=2$, $thres_2=4$. En este ejemplo, contenedor0 y contenedor1 comparten los mismos contextos y contenedor2 y contenedor3 comparten los mismos contextos.

15 El indicador de interpredicción (*inter_pred_flag*) especifica si se usa predicción única o predicción doble para la PU actual. En algunos ejemplos, el índice de contexto para el indicador de interpredicción es igual a la profundidad de CU actual. Como hay cuatro profundidades de CU posibles (0 a 3), hay cuatro contextos posibles para codificar el *inter_pred_flag*.

20 Esta divulgación propone que el índice de contexto usado para seleccionar un contexto para codificar el *inter_pred_flag* es igual a la profundidad de CU actual (por ejemplo, la descomposición de árbol cuaternario de nivel para CU), pero está limitado a un umbral elegido (es decir, es el menor de la profundidad de CU actual o un umbral). En un ejemplo, el umbral puede elegirse como 2. Alternativamente, el índice de contexto puede ser igual a la profundidad máxima de CU menos la profundidad de CU actual y limitarse a un umbral elegido. Alternativamente, puede diseñarse una tabla de mapeo predefinida para seleccionar el índice de contexto según una profundidad de CU determinada. La tabla de mapeo puede implementarse como un conjunto de lógica. Como resultado, se usan 3 contextos para codificar el elemento de sintaxis *inter_pred_flag*.

25 La Tabla 11 muestra el rendimiento de codificación cuando se cambia la tabla de inicialización, pero no se modifica el número de contextos. La Tabla 12 muestra el rendimiento de codificación de la técnica propuesta que reduce el número de contextos de 4 a 3.

30 Tabla 11. Rendimiento de codificación de HM4.0 con inicialización de CABAC modificada en *inter_pred_flag*.

Tasa de BD	Todo Intra HE			Acceso aleatorio HE			B de bajo retardo HE		
	Y	U	V	Y	U	V	Y	U	V
Clase A				0,03 %	-0,15 %	-0,11 %			
Clase B				-0,03 %	0,01 %	-0,03 %	0,03 %	-0,02 %	-0,11 %
Clase C				0,00 %	-0,12 %	0,06 %	-0,03 %	-0,16 %	0,01 %
Clase D				-0,01 %	-0,04 %	0,01 %	-0,09 %	0,51 %	0,20 %
Clase E							0,10 %	-0,03 %	0,65 %
Todo				-0,01 %	-0,07 %	-0,02 %	0,00 %	0,07 %	0,14 %
Enc T[%]									
Dec T[%]									

45 Tabla 12. Rendimiento de codificación de la técnica de reducción de contexto propuesta en *inter_pred_flag*.

Tasa de BD	Todo Intra HE			Acceso aleatorio HE			B de bajo retardo HE		
	Y	U	V	Y	U	V	Y	U	V
Clase A				0,05 %	-0,11 %	-0,14 %			
Clase B				-0,01 %	-0,03 %	0,02 %	0,00 %	-0,01 %	-0,15 %
Clase C				-0,02 %	-0,14 %	-0,02 %	0,01 %	0,01 %	-0,03 %
Clase D				0,03 %	-0,01 %	-0,01 %	-0,09 %	-0,12 %	-0,01 %
Clase E							-0,12 %	0,08 %	0,45 %
Todo				0,01 %	-0,07 %	-0,03 %	-0,04 %	-0,01 %	0,03 %
Enc T[%]									
Dec T[%]									

55 El índice del cuadro de referencia (*ref_idx_lx*) se señala mediante el uso de un código unario truncado con respecto al cuadro de referencia activo en la lista asociada (por ejemplo, Lista 0 o Lista 1). Se usan tres contextos para codificar el índice del cuadro de referencia. Un contexto para el contenedor 0, un contexto para el contenedor 1 y se usa un contexto en el resto de los contenedores. La Tabla 13 muestra un ejemplo de asignaciones de contexto para los contenedores del código unario para *ref_idx_lx*.

65

Tabla 13. Asignación de contexto para contenedores de *ref_idx_lx*

Contenedor de código unario <i>ref_idx_lx</i>	Contexto
Contenedor 0	Contexto 1
Contenedor 1	Contexto 2
Contenedores 2-N (N es el número total de contenedores)	Contexto 3

Esta divulgación propone usar dos contextos para codificar el código unario para *ref_idx_lx*; un contexto en el contenedor 0 y otro contexto en el resto de los contenedores. La Tabla 14 muestra un ejemplo de asignación de contexto para los contenedores del código unario para *ref_idx_lx* de acuerdo con este ejemplo de la divulgación. La Tabla 15 muestra el rendimiento de codificación asociado a las modificaciones propuestas.

Tabla 14. Asignación de contexto para contenedores de *ref_idx_lx*

Contenedor de código unario <i>ref_idx_lx</i>	Contexto
Contenedor 0	Contexto 1
Contenedores 1-N (N es el número total de contenedores)	Contexto 2

Tabla 15. Rendimiento de codificación del procedimiento propuesto en *ref_idx_lx*.

Tasa de BD	Todo Intra HE			Acceso aleatorio HE			B de bajo retardo HE		
	Y	U	V	Y	U	V	Y	U	V
Clase A				-0,01 %	-0,11 %	-0,16 %			
Clase B				-0,01 %	0,00 %	-0,01 %	0,01 %	-0,12 %	0,01 %
Clase C				-0,01 %	0,02 %		-0,04 %	-0,14 %	-0,07 %
Clase D				0,03 %	0,06 %	0,11 %	-0,06 %	0,19 %	-0,09 %
Clase E							-0,06 %	-0,34 %	0,48 %
Todo				0,00 %	-0,01 %	-0,01 %	-0,03 %	-0,09 %	0,05 %
Enc T[%]									
Dec T[%]									

Para los elementos de sintaxis del indicador de bloque codificado por croma (*cbf_cb* y *cbf_cr*), se usan dos conjuntos de contexto diferentes (5 contextos en cada conjunto de contexto) para CABAC. El índice del contexto real usado en cada conjunto es igual a la profundidad de transformación actual asociada con el indicador de bloque codificado por croma que se está codificando. La Tabla 16 muestra los conjuntos de contexto para los indicadores de bloque codificado por croma *cbf_cb* y *cbf_cr*.

Tabla 16. Conjuntos de contexto para *cbf_cb* y *cbf_cr*

Indicador de bloque codificado por croma	Conjunto de contexto
<i>Cbf_cb</i>	Conjunto de contexto 1 (5 contextos)
<i>Cbf_cr</i>	Conjunto de contexto 2 (5 contextos)

Esta divulgación propone que *cbf_cb* y *cbf_cr* compartan un conjunto de contexto. El índice del contexto real usado en cada conjunto aún puede ser igual a la profundidad de transformación actual asociada con el indicador de bloque codificado por croma que se está codificando. La Tabla 17 muestra los conjuntos de contexto para los indicadores de bloque codificado por croma *cbf_cb* y *cbf_cr* de acuerdo con ejemplos de esta divulgación. La Tabla 18 muestra el rendimiento de codificación asociado a las modificaciones propuestas.

Tabla 17. Conjuntos de contexto para *cbf_cb* y *cbf_cr* de acuerdo con ejemplos de esta divulgación

Indicador de bloque codificado por croma	Conjunto de contexto
<i>Cbf_cb</i>	Conjunto de contexto 1 (5 contextos)
<i>Cbf_cr</i>	Conjunto de contexto 1 (5 contextos)

Tabla 18. Rendimiento de codificación del procedimiento propuesto en *cbf_cb* y *cbf_cr*.

Tasa de BD	Todo Intra HE			Acceso aleatorio HE			B de bajo retardo HE		
	Y	U	V	Y	U	V	Y	U	V
Clase A	-0,01 %	0,59 %	-1,06 %	0,02 %	0,56 %	-1,71 %	0,01 %	1,06 %	-1,89 %
Clase B	0,00 %	0,59 %	-1,07 %	-0,01 %	0,68 %	-1,32 %	-0,01 %	0,21 %	-0,97 %
Clase C	-0,01 %	0,17 %	-0,75 %	0,00 %	0,09 %	-0,63 %	0,04 %	-0,36 %	-0,45 %
Clase D	0,00 %	-0,17 %	-0,51 %	0,04 %	-0,23 %	-0,80 %	0,04 %	0,36 %	-0,40 %
Clase E	0,00 %	-0,36 %	0,24 %						
Todo	0,00 %	0,21 %	-0,70 %	0,01 %	0,30 %	-1,13 %	0,02 %	0,36 %	-0,87 %
Enc T[%]									
Dec T[%]									

De esta manera, de acuerdo con este ejemplo, tanto el codificador de vídeo 20 como el decodificador de vídeo 30 pueden configurarse para codificar un indicador de bloque codificado por croma Cb para un bloque de datos de vídeo mediante el uso de la codificación aritmética binaria adaptativa al contexto (CABAC), en el que CABAC usa un conjunto de contexto que incluye uno o más contextos, y codificar un indicador de bloque codificado por croma Cr mediante el uso de CABAC, en el que CABAC usa el mismo conjunto de contexto que el indicador de bloque codificado por croma Cb. El codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse además para seleccionar un contexto de uno o más contextos en base a una profundidad de transformación de una unidad de transformación asociada con el bloque de datos de vídeo.

En una propuesta para HEVC, hay doce conjuntos de contexto tanto para *coeff_abs_level_greater1_flag* como para *coeff_abs_level_greater2_flag*. El *coeff_abs_level_greater1_flag* indica si un coeficiente de transformación tiene un valor absoluto mayor que 1. El *coeff_abs_level_greater2_flag* indica si un coeficiente de transformación tiene un valor absoluto mayor que 2. Los conjuntos de contexto se asignan por igual para los componentes de luma y croma, es decir, 6 conjuntos de contexto para luma y 6 contextos para croma. Cada conjunto de contexto consta de 5 contextos. El índice del conjunto de contexto, *ctxSet*, se selecciona en base al anterior *coeff_abs_level_greater_flag*. Para *coeff_abs_level_greater1_flag*, el índice del contexto dentro de un conjunto de contexto, *greater1Ctx*, se determina en base a los últimos hasta un máximo de 4. El índice de contexto puede representarse como:

$$\text{ctxIdx_level_greater1} = (\text{ctxSet} * 5) + \text{Min}(4, \text{greater1Ctx}) \quad (1)$$

Para *coeff_abs_level_greater2_flag*, el índice del contexto dentro de un conjunto de contexto, *greater2Ctx*, es en base al número de *coeff_abs_level_greater1_flag* siendo de 1 a un máximo de 4. El índice de contexto puede representarse como:

$$\text{ctxIdx_level_greater2} = (\text{ctxSet} * 5) + \text{Min}(4, \text{greater2Ctx}) \quad (2)$$

greater1Ctx en base al número de coeficientes significativos y el número de coeficientes que son mayores que 1. Por otro lado, *greater2Ctx* es en base al número de coeficientes que son mayores que 1.

En algunos ejemplos, puede usarse un número diferente de contextos en diferentes conjuntos de contexto, incluidos, por ejemplo:

1. Los conjuntos de contexto para un nivel mayor que 1 o para un nivel mayor que 2 podrían tener un número diferente de contextos. Por ejemplo, el conjunto de contexto 0 y 3 podrían tener 5 contextos y el resto de los conjuntos de contexto podrían tener 2 contextos.
2. Los conjuntos de contexto para el coeficiente de luma pueden tener un número diferente de contextos en comparación con los conjuntos de contexto para el componente de croma. Por ejemplo, el conjunto de contexto 0 para luma puede tener 5 contextos y el conjunto de contexto 0 para croma podría tener 4 contextos.
3. El conjunto de contexto para un nivel mayor que 1 puede tener un número de contextos diferente al del conjunto de contexto para un nivel mayor que 2. Por ejemplo, el conjunto de contexto 0 para un nivel mayor que 1 podría tener 5 contextos y el conjunto de contexto 0 para un nivel mayor que 2 solo podría tener 3 contextos.

En otros ejemplos, puede usarse un número diferente para los conjuntos de contexto para la codificación de mayor que 1 o mayor que 2, incluidos, por ejemplo:

1. Los conjuntos de contexto para el coeficiente de luma pueden tener un número diferente de conjuntos de contexto a los conjuntos de contexto usados para el componente de croma. Por ejemplo, la luma podría usar 6 contextos y el croma podría usar 4 contextos.
2. Los conjuntos de contexto mayores que 1 pueden tener un número diferente de conjuntos de contexto para los conjuntos de contexto usados mayores que 2. Por ejemplo, el mayor que 1 podría usar 6 contextos y el mayor que 2 podría usar 4 contextos.

Opcionalmente, se usa una métrica para determinar qué contexto se está usando en un conjunto de contexto y el intervalo de valores de la métrica es mayor que el número de contextos en el conjunto de contexto. En uno de estos aspectos, un contexto podría asociarse a uno o más valores de la métrica. El intercambio de contexto se limita preferentemente a valores continuos. Por ejemplo, sea $y=2$ se asocia al contexto 3, e $y=1$ e $y=4$ también pueden asociarse al contexto 3. Sin embargo, si $y=3$ se asocia al contexto 4, $y=4$ no puede asociarse al contexto 3.

Por ejemplo, para *coeff_abs_level_greater1_flag*, el conjunto de contexto 0 y 3 tienen 5 contextos y el conjunto de contexto 1, 2, 4 y 5 tienen 2 contextos. Para *coeff_abs_level_greater2_flag*, el conjunto de contexto 0, 1 y 2 tienen 5 contextos y el conjunto de contexto 3, 4 y 5 tienen 2 contextos. Puede representarse como:

$$\text{ctxIdx_level_greater1} = (\text{ctxSet} * 5) + \text{Min}(\text{Thres_greater1}, \text{greater1Ctx}) \quad (3)$$

si $\text{ctxSet} = 0$ o $\text{ctxSet} = 3$, $\text{Thres_greater1} = 4$;
de cualquier otra manera, $\text{Thres_greater1} = 1$.

$$\text{ctxIdx_level_greater2} = (\text{ctxSet} * 5) + \text{Min}(\text{Thres_greater2}, \text{greater2Ctx}) \quad (4)$$

si $\text{ctxSet} < 3$, $\text{Thres_greater2} = 4$;
de cualquier otra manera, $\text{Thres_greater2} = 1$

Thres_greater1 y Thres_greater2 pueden elegirse de forma diferente en base a las siguientes situaciones:

1. Componente de luma o croma
2. Conjuntos de contexto

Como otro ejemplo, para *coeff_abs_level_greater1_flag*, el conjunto de contexto 0 y 3 tienen 5 contextos y el conjunto de contexto 1, 2, 4 y 5 tienen 3 contextos. Para *coeff_abs_level_greater2_flag*, el conjunto de contexto 0, 1 y 2 tienen 5 contextos y el conjunto de contexto 3, 4 y 5 tienen 2 contextos. Puede representarse como:

$$\text{ctxIdx_level_greater1} = (\text{ctxSet} * 5) + \text{greater1Ctx_mapped} \quad (3)$$

$$\text{ctxIdx_level_greater2} = (\text{ctxSet} * 5) + \text{greater2Ctx_mapped} \quad (4)$$

En tales ejemplos, el mapeo puede ser como se muestra en las Tablas 19 y 20:

Tabla 19

greater1Ctx	0	1	2	3	>3
ctxSet 0	0	1	2	3	4
ctxSet 1	0	1	1	2	2
ctxSet 2	0	1	1	1	2
ctxSet 3	0	1	2	3	4
ctxSet 4	0	1	2	2	2
ctxSet 5	0	1	1	2	2

Tabla 20

greater2Ctx	0	1	2	3	>3
ctxSet 0	0	1	2	3	4
ctxSet 1	0	1	1	1	1
ctxSet 2	0	1	1	1	1
ctxSet 3	0	1	2	3	4
ctxSet 4	0	1	1	1	1
ctxSet 5	0	1	1	1	1

Las tablas de inicialización de CABAC de *coeff_abs_level_greater1_flag* y *coeff_abs_level_greater2_flag* también se modifican para los conjuntos de contexto para Thres_greater1 o Thres_greater2 iguales a 1. Las modificaciones adelantan la inicialización del quinto contexto para que sea la inicialización del segundo contexto. Este procedimiento propuesto reduce el número de contextos de 120 a 78.

Tabla 21. Rendimiento de codificación del procedimiento propuesto en *coeff_abs_level_greater1_flag* y *coeff_abs_level_greater2_flag*.

Tasa de BD	Todo Intra HE			Acceso aleatorio HE			B de bajo retardo HE		
	Y	U	V	Y	U	V	Y	U	V
Clase A	0,00 %	-0,04 %	0,03 %	0,05 %	0,31 %	-0,35 %			
Clase B	0,01 %	0,04 %	0,03 %	0,01 %	0,03 %	-0,09 %	0,00 %	-0,15 %	-0,23 %
Clase C	0,00 %	0,05 %	0,00 %	0,03 %	0,06 %	0,06 %	0,00 %	0,23 %	-0,23 %
Clase D	0,00 %	0,01 %	-0,03 %	0,01 %	0,22 %	0,04 %	-0,01 %	0,26 %	0,24 %
Clase E	0,00 %	-0,02 %	0,03 %				0,09 %	-0,52 %	0,16 %
Todo	0,00 %	0,01 %	0,01 %	0,02 %	0,15 %	-0,09 %	0,01 %	-0,02 %	-0,04 %
Enc T[%]									
Dec T[%]									

En la Tabla 21 se enumera el número de contextos para todos los elementos de sintaxis mencionados en secciones anteriores. La reducción total es de 56 contextos.

Tabla 22. Comparación del número de contextos en el procedimiento propuesto y HM4.0

# de Contextos	HM4.0	Procedimiento propuesto
pred_type	10	6
merge_idx	4	1
inter_pred_flag	4	3
ref_idx_lc, ref_idx_10, ref_idx_l1	3	2
cbf_cb, cbf_cr	10	5
coeff_abs_level_greater1_flag	60	36
coeff_abs_level_greater2_flag	60	42
Total	151	95

La Figura 2 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo que puede implementar las técnicas descritas en la presente divulgación. El codificador de vídeo puede realizar la intracodificación y la intercodificación de los bloques de vídeo dentro de los segmentos de vídeo. La intracodificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial en el vídeo dentro de un cuadro o imagen de vídeo determinado. La intercodificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal en el vídeo dentro de cuadros o imágenes adyacentes de una secuencia de vídeo. El intramodo (modo I) puede referirse a cualquiera de diversos modos de compresión basados en el espacio. Los intermodos, tal como la predicción unidireccional (modo P) o la predicción bidireccional (modo B), se pueden referir a cualquiera de diversos modos de compresión basados en el tiempo.

En el ejemplo de la Figura 2, el codificador de vídeo 20 incluye una unidad de partición 35, una unidad de predicción 41, una memoria de imágenes de referencia 64, un sumador 50, una unidad de transformación 52, una unidad de cuantificación 54 y una unidad de codificación por entropía 56. La unidad de predicción 41 incluye la unidad de estimación del movimiento 42, la unidad de compensación del movimiento 44 y la unidad de intrapredicción 46. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 también incluye la unidad de cuantificación inversa 58, la unidad de transformación inversa 60, y el sumador 62. También puede incluirse un filtro de desbloqueo (no mostrado en la Figura 2) para filtrar los límites de bloque y eliminar los artefactos de bloqueo del vídeo reconstruido. Si se desea, el filtro de desbloqueo típicamente filtraría la salida del sumador 62. Además del filtro de desbloqueo, pueden usarse otros filtros de bucle (en bucle o postbucle).

Como se muestra en la Figura 2, el codificador de vídeo 20 recibe los datos de vídeo, y la unidad de partición 35 particiona los datos en los bloques de vídeo. Esta partición también puede incluir la partición en segmentos, mosaicos u otras unidades más grandes, así como también la partición de bloques de vídeo, por ejemplo, de acuerdo con una estructura de cuatro árboles de LCU y CU. El codificador de vídeo 20 generalmente ilustra los componentes que codifican bloques de vídeo dentro de un segmento de vídeo a codificar. El segmento puede dividirse en múltiples bloques de vídeo (y posiblemente en conjuntos de bloques de vídeo denominados mosaicos). La unidad de predicción 41 puede seleccionar uno de una pluralidad de modos de codificación posibles, tal como uno de una pluralidad de modos de codificación intra o uno de una pluralidad de modos de codificación inter, para el bloque de vídeo actual en base a los resultados de error (por ejemplo, la tasa de codificación y el nivel de distorsión). La unidad de predicción 41 puede proporcionar el bloque intra o intercodificado resultante al sumador 50 para generar los datos del bloque residual y al sumador 62 para reconstruir el bloque codificado para usarlo como una imagen de referencia.

La unidad de intrapredicción 46 dentro de la unidad de predicción 41 puede realizar la codificación intrapredictiva del

bloque de vídeo actual en relación con uno o más bloques vecinos en el mismo cuadro o segmento que el bloque actual a codificar para proporcionar compresión espacial. La unidad de estimación del movimiento 42 y la unidad de compensación del movimiento 44 dentro de la unidad de predicción 41 realizan la codificación interpredictiva del bloque de vídeo actual en relación con uno o más bloques predictivos en una o más imágenes de referencia para proporcionar compresión temporal.

La unidad de estimación del movimiento 42 puede configurarse para determinar el modo de interpredicción para un segmento de vídeo de acuerdo con un patrón predeterminado para una secuencia de vídeo. El patrón predeterminado puede designar segmentos de vídeo en la secuencia como segmento P, segmento B o segmentos GPB (P/B generalizado). La unidad de estimación del movimiento 42 y la unidad de compensación del movimiento 44 pueden estar muy integradas, pero se ilustran por separado con fines conceptuales. La estimación de movimiento, realizada por la unidad de estimación del movimiento 42, es el procedimiento de generar los vectores de movimiento, que estiman el movimiento para los bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de una PU de un bloque de vídeo dentro de una imagen o cuadro de vídeo actual con relación a un bloque predictivo dentro de una imagen de referencia.

Un bloque predictivo es un bloque que se encuentra que coincide estrechamente con la PU del bloque de vídeo a codificar en términos de diferencia de píxeles, que se puede determinar mediante la suma de la diferencia absoluta (SAD), la suma de la diferencia cuadrada (SSD) o mediante otras métricas de diferencia. En algunos ejemplos, el codificador de vídeo 20 puede calcular valores para posiciones de píxeles subenteros de imágenes de referencia almacenadas en la memoria de imágenes de referencia 64. Por ejemplo, el codificador de vídeo 20 puede interpolar valores de posiciones de un cuarto de píxel, posiciones de un octavo de píxel u otras posiciones de píxeles fraccionarios de la imagen de referencia. Por lo tanto, la unidad de estimación del movimiento 42 puede realizar una búsqueda de movimiento en relación con las posiciones de píxeles completos y a las posiciones de píxeles fraccionarios y generar un vector de movimiento con precisión de píxeles fraccionarios.

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

La compensación de movimiento, realizada por la unidad de compensación del movimiento 44, puede implicar buscar o generar el bloque predictivo en base al vector de movimiento determinado por la estimación de movimiento, posiblemente realizando interpolaciones con precisión de subpíxeles. Al recibir el vector de movimiento para la PU del bloque de vídeo actual, la unidad de compensación del 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 vídeo 20 forma un bloque de vídeo residual restando los valores de píxeles del bloque predictivo de los valores de píxeles del bloque de vídeo actual que se codifica, formando valores de diferencia de píxeles. Los valores de diferencia de píxeles forman los datos residuales para el bloque y pueden incluir componentes de diferencia de luma y croma. El sumador 50 representa el componente o componentes que realizan esta operación de sustracción. La unidad de compensación del movimiento 44 también puede generar los elementos de sintaxis asociados con los bloques de vídeo y el segmento de vídeo para su uso por el decodificador de vídeo 30 al decodificar los bloques de vídeo del segmento de vídeo.

La unidad de intrapredicción 46 puede intrapredicir un bloque actual, como una alternativa a la interpredicción realizada por la unidad de estimación del movimiento 42 y la unidad de compensación del movimiento 44, como se describió anteriormente. En particular, la unidad de intrapredicción 46 puede determinar un modo de intrapredicción a usar para codificar un bloque actual. En algunos ejemplos, la unidad de intrapredicción 46 puede codificar un bloque actual mediante el uso de diversos modos de intrapredicción, por ejemplo, durante pasadas de codificación separadas, y la unidad de intrapredicción 46 (o la unidad de selección de modo 40, en algunos ejemplos) puede seleccionar un modo de intrapredicción apropiado para usar entre los modos probados. Por ejemplo, la unidad de intrapredicción 46 puede calcular los valores de tasa de distorsión mediante el uso de un análisis de la tasa de distorsión para los diversos modos de intrapredicción probados, y seleccionar el modo de intrapredicción que tiene las mejores características de tasa de distorsión entre los modos probados. El análisis de la tasa de distorsión generalmente determina una cantidad de distorsión (o error) entre un bloque codificado y uno original, el bloque no codificado que se codificó para producir el bloque codificado, así como también una tasa de bits (es decir, una cantidad de bits) que se usa para producir el bloque codificado. La unidad de intrapredicción 46 puede calcular las relaciones de las distorsiones y tasas para diversos bloques codificados para determinar qué modo de intrapredicción exhibe el mejor valor de tasa de distorsión para el bloque.

En cualquier caso, después de seleccionar un modo de intrapredicción para un bloque, la unidad de intrapredicción 46 puede proporcionar información indicativa del modo de intrapredicción seleccionado para el bloque a la unidad de codificación por entropía 56. La unidad de codificación por entropía 56 puede codificar la información que indica el

modo de intrapredicción seleccionado de acuerdo con las técnicas de esta divulgación. El codificador de vídeo 20 puede incluir en los datos de configuración del flujo de bits transmitido, que pueden incluir una pluralidad de tablas de índice en modo de intrapredicción y una pluralidad de tablas de índice en modo de intrapredicción modificadas (que se refieren además, a tablas de mapeo de palabras de código), definiciones de contextos de codificación para diversos bloques e indicaciones de un modo de intrapredicción más probable, una tabla de índice de modo de intrapredicción y una tabla de índice de modo de intrapredicción modificada para usar en cada uno de los contextos.

Después de que la unidad de predicción 41 genere el bloque predictivo para el bloque de vídeo actual a través de interpredicción o intrapredicción, el codificador de vídeo 20 forma un bloque de vídeo residual al restar el bloque predictivo del bloque de vídeo actual. Los datos de vídeo residuales del bloque residual pueden incluirse en una o más UT y aplicarse a la unidad de transformación 52. La unidad de transformación 52 transforma los datos de vídeo residuales en coeficientes de transformación residuales mediante el uso de una transformada, tal como una transformada coseno discreta (DCT) o una transformada conceptualmente similar. La unidad de transformación 52 puede convertir los datos de vídeo residuales de un dominio de píxeles a un dominio de transformación, tal como un dominio de frecuencia.

La unidad de transformación 52 puede enviar los coeficientes de transformación resultantes a la unidad de cuantificación 54. La unidad de cuantificación 54 cuantifica los coeficientes de transformación para reducir además la tasa de bits. El procedimiento de cuantificación puede reducir la profundidad de bits asociada con algunos o todos los coeficientes. El grado de cuantificación puede modificarse mediante el ajuste de 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 por entropía 56 puede realizar la exploración. A modo de ejemplo, las técnicas de codificación descritas en esta divulgación pueden realizarse total o parcialmente por la unidad de codificación por entropía 56. Sin embargo, algunos aspectos de esta divulgación no se limitan. Por ejemplo, las técnicas de codificación descritas en esta divulgación pueden realizarse por un componente del codificador de vídeo 20 no mostrado en la Figura 2, tal como un procesador o cualquier otro componente. En algunos ejemplos, las técnicas de codificación de esta divulgación pueden realizarse por una de las otras unidades o módulos ilustrados en la Figura 2. En algunos otros ejemplos más, las técnicas de codificación de esta divulgación pueden realizarse mediante una combinación de unidades y módulos del codificador de vídeo 20. De esta manera, el codificador de vídeo 20 puede configurarse para realizar las técnicas de ejemplo descritas en esta divulgación.

Después de la cuantificación, la entropía de la unidad de codificación por entropía 56 codifica los coeficientes de transformación cuantificados. Por ejemplo, la unidad de codificación por 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 en base a sintaxis (SBAC), codificación por entropía de partición de intervalo de probabilidad (PIPE) u otra metodología o técnica de codificación por entropía. Después de la codificación por entropía mediante la unidad de codificación por entropía 56, el flujo de bits codificado se puede transmitir al decodificador de vídeo 30 o archivar para su posterior transmisión o recuperación mediante el decodificador de vídeo 30. La unidad de codificación por entropía 56 también puede codificar por entropía los vectores de movimiento y los otros elementos de sintaxis para el segmento de vídeo actual que se codifica.

En un ejemplo de la divulgación, la unidad de codificación por entropía 56 puede configurarse para determinar un primer tipo de predicción para un bloque de datos de vídeo en un segmento P, representar el primer tipo de predicción como un elemento de sintaxis de tipo de predicción de segmento P, determinar un segundo tipo de predicción para un bloque de datos de vídeo en un segmento B, representar el segundo tipo de predicción como un elemento de sintaxis de tipo de predicción de segmento B, determinar una binarización de segmento P para el elemento de sintaxis de tipo de predicción de segmento P, determinar una binarización de segmento B para el elemento de sintaxis de tipo de predicción de segmento B, en el que el elemento de sintaxis de tipo de predicción de segmento P y el elemento de sintaxis de tipo de predicción de segmento B se determinan mediante el uso de la misma lógica de binarización, y codificar los datos de vídeo en base a las binarizaciones del elemento de sintaxis de tipo de predicción de segmento P y el elemento de sintaxis de predicción de segmento B.

En otro ejemplo de la divulgación, la unidad de codificación por entropía 56 puede configurarse para determinar un tipo de partición para un modo de predicción para un bloque de datos de vídeo, codificar un contenedor de tipo de partición de un elemento de sintaxis de tipo de predicción para el bloque de datos de vídeo mediante el uso de la codificación aritmética binaria adaptativa al contexto con un único contexto, en el que el único contexto es el mismo para cualquier tipo de partición, y codificar un contenedor de tamaño de partición de una sintaxis de tipo de predicción para el bloque de datos de vídeo mediante el uso de la codificación aritmética binaria adaptativa al contexto en modo derivación.

En otro ejemplo de la divulgación, la unidad de codificación por entropía 56 puede configurarse para codificar un indicador de bloque codificado por croma Cb para un bloque de datos de vídeo mediante el uso de la codificación aritmética binaria adaptativa al contexto (CABAC), en el que CABAC usa un conjunto de contexto que incluye uno o más contextos, y codificar un indicador de bloque codificado por croma Cr mediante el uso de CABAC, en el que CABAC usa el mismo conjunto de contexto que el indicador de bloque codificado por croma Cb. El codificador de

vídeo 20 y el decodificador de vídeo 30 pueden configurarse además para seleccionar un contexto de uno o más contextos en base a una profundidad de transformación de una unidad de transformación asociada con el bloque de datos de vídeo.

La unidad de cuantificación inversa 58 y la unidad de transformación inversa 60 aplican la cuantificación inversa y la transformación inversa, respectivamente, para reconstruir el bloque residual en el dominio del píxel para su uso posterior como un bloque de referencia de una imagen de referencia. La unidad de compensación del movimiento 44 puede calcular un bloque de referencia añadiendo 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 del movimiento 44 también puede aplicar uno o más filtros de interpolación al bloque residual reconstruido para calcular valores de píxeles subenteros para su uso en la estimación de movimiento. El sumador 62 añade el bloque residual reconstruido al bloque de predicción compensado de movimiento producido por la unidad de compensación del movimiento 44 para producir un bloque de referencia para su almacenamiento en la memoria de imágenes de referencia 64. El bloque de referencia se puede utilizar por la unidad de estimación del movimiento 42 y por la unidad de compensación del movimiento 44 como un bloque de referencia para interpredecir un bloque en un cuadro o imagen de vídeo subsiguiente.

La Figura 3 es un diagrama de bloques que ilustra un ejemplo de decodificador de vídeo 30 que puede implementar las técnicas descritas en la presente divulgación. En el ejemplo de la Figura 3, el decodificador de vídeo 30 incluye una unidad de decodificación por entropía 80, una unidad de predicción 81, una unidad de cuantificación inversa 86, una unidad de transformación inversa 88, un sumador 90 y una memoria de imágenes de referencia 92. La unidad de predicción 81 incluye la unidad de compensación del movimiento 82 y la unidad de intrapredicción 84. El decodificador de vídeo 30 puede, en algunos ejemplos, realizar una pasada de decodificación generalmente recíproca a la pasada de codificación descrita con respecto al codificador de vídeo 20 de la Figura 2.

Durante el procedimiento de decodificación, el decodificador de vídeo 30 recibe un flujo de bits de vídeo codificado que representa los bloques de vídeo de un segmento de vídeo codificado y los elementos de sintaxis asociados del codificador de vídeo 20. La unidad de decodificación por entropía 80 del decodificador de vídeo 30 decodifica entrópicamente el flujo de bits para generar coeficientes cuantificados, vectores de movimiento y otros elementos de sintaxis. La unidad de decodificación por entropía 80 envía los vectores de movimiento y otros elementos de sintaxis a la unidad de predicción 81. El decodificador de vídeo 30 puede recibir los elementos de sintaxis a nivel de segmento de vídeo y/o a nivel de bloque de vídeo.

A modo de ejemplo, las técnicas de codificación descritas en esta divulgación pueden realizarse total o parcialmente por la unidad de decodificación por entropía 80. Sin embargo, algunos aspectos de esta divulgación no se limitan. Por ejemplo, las técnicas de codificación descritas en esta divulgación pueden realizarse por un componente del decodificador de vídeo 30 no mostrado en la Figura 3, tal como un procesador o cualquier otro componente. En algunos ejemplos, las técnicas de codificación de esta divulgación pueden realizarse por una de las otras unidades o módulos ilustrados en la Figura 3. En algunos otros ejemplos más, las técnicas de codificación de esta divulgación pueden realizarse mediante una combinación de unidades y módulos del decodificador de vídeo 30. De esta manera, el decodificador de vídeo 30 puede configurarse para realizar las técnicas de ejemplo descritas en esta divulgación.

En un ejemplo de la divulgación, la unidad de decodificación por entropía 80 puede configurarse para mapear un elemento de sintaxis de tipo de predicción de segmento P binarizado a un tipo de predicción mediante el uso de un mapeo de binarización para un bloque de datos de vídeo en un segmento P, mapear un elemento de sintaxis de tipo de predicción de segmento B binarizado a un tipo de predicción mediante el uso del mismo mapeo de binarización para un bloque de datos de vídeo en un segmento B, y decodificar los datos de vídeo en base a los tipos de predicción mapeados.

En un ejemplo de la divulgación, la unidad de decodificación por entropía 80 puede configurarse para recibir un elemento de sintaxis de tipo de predicción para un bloque de datos de vídeo que ha sido codificado mediante el uso de la codificación aritmética binaria adaptativa al contexto (CABAC), incluyendo el elemento de sintaxis de tipo de predicción de un contenedor de tipo de partición que representa un tipo de partición y un contenedor de tamaño de partición que representa un tamaño de partición, decodificando el contenedor de tipo de partición del elemento de sintaxis de tipo de predicción mediante el uso de la codificación aritmética binaria adaptativa al contexto con un único contexto, en el que el único contexto es el mismo para cualquier tipo de partición, y decodificar el contenedor de tamaño de partición de la sintaxis de tipo de predicción para un bloque de datos de vídeo mediante el uso de la codificación aritmética binaria adaptativa al contexto en modo derivación.

En otro ejemplo de la divulgación, la unidad de decodificación por entropía 80 puede configurarse para codificar un indicador de bloque codificado por croma Cb para un bloque de datos de vídeo mediante el uso de la codificación aritmética binaria adaptativa al contexto (CABAC), en el que CABAC usa un conjunto de contexto que incluye uno o más contextos, y codificar un indicador de bloque codificado por croma Cr mediante el uso de CABAC, en el que CABAC usa el mismo conjunto de contexto que el indicador de bloque codificado por croma Cb. El codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse además para seleccionar un contexto de uno o más

contextos en base a una profundidad de transformación de una unidad de transformación asociada con el bloque de datos de vídeo.

5 Cuando el segmento de vídeo se codifica como un segmento intracodificado (I), la unidad de intrapredicción 84 de la unidad de predicción 81 puede generar los datos de predicción para un bloque de vídeo del segmento de vídeo actual en base a un modo de intrapredicción señalizado y en los datos de bloques previamente decodificados del cuadro o imagen actual. Cuando el cuadro de vídeo se codifica como un segmento intercodificado (es decir, B, P o GPB), la unidad de compensación del movimiento 82 de la unidad de predicción 81 produce los bloques predictivos para un bloque de vídeo del segmento de vídeo actual en base a los vectores de movimiento y otros elementos de sintaxis recibidos de la unidad de decodificación por entropía 80. Los bloques predictivos pueden producirse a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El decodificador de vídeo 30 puede construir listas de cuadros de referencia, Lista 0 y Lista 1, mediante el uso de técnicas de construcción predeterminadas en base a las imágenes de referencia almacenadas en la memoria de imágenes de referencia 92.

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

25 La unidad de compensación del movimiento 82 también puede realizar una interpolación en base a filtros de interpolación. La unidad de compensación del movimiento 82 puede usar filtros de interpolación como los usa el codificador de vídeo 20 durante la codificación de los bloques de vídeo para calcular valores interpolados para píxeles subenteros de bloques de referencia. En este caso, la unidad de compensación del movimiento 82 puede determinar los filtros de interpolación usados por el codificador de vídeo 20 a partir de los elementos de sintaxis recibidos y usar los filtros de interpolación para producir bloques predictivos.

35 La unidad de cuantificación inversa 86 cuantifica inversamente, es decir, descuantifica, los coeficientes de transformación cuantificados proporcionados en el flujo de bits y decodificados por la unidad de decodificación por entropía 80. El procedimiento de cuantificación inversa puede incluir el uso de un parámetro de cuantificación calculado por el codificador de vídeo 20 para cada bloque de vídeo en el segmento de vídeo para determinar un grado de cuantificación e, igualmente, un grado de cuantificación inversa que debería aplicarse. La unidad de transformación inversa 88 aplica una transformada inversa, por ejemplo, una DCT inversa, una transformada de entero inverso o un procedimiento de transformada inversa conceptualmente similar, a los coeficientes de transformación con el fin de producir bloques residuales en el dominio de píxeles.

45 Después de que la unidad de compensación del movimiento 82 genere el bloque predictivo para el bloque de vídeo actual en base a los vectores de movimiento y otros elementos de sintaxis, el decodificador de vídeo 30 forma un bloque de vídeo decodificado al sumar los bloques residuales de la unidad de transformación inversa 88 con los correspondientes bloques predictivos generados por la unidad de compensación del movimiento 82. El sumador 90 representa el componente o componentes que realizan esta operación de suma. Si se desea, también puede aplicarse un filtro de desbloqueo para filtrar los bloques decodificados con el fin de eliminar los artefactos de bloqueo. Pueden además, usarse otros filtros de bucle (ya sea en el bucle de codificación o después del bucle de codificación) para suavizar las transiciones de píxeles o mejorar de cualquier otro modo la calidad del vídeo. Los bloques de vídeo decodificados en un cuadro o imagen dadas se almacenan luego en la memoria de imágenes de referencia 92, que almacena las imágenes de referencia usadas para la compensación de movimiento subsiguiente. La memoria de imágenes de referencia 92 también almacena vídeo decodificado para su posterior presentación en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la Figura 1.

55 La Figura 6 es un diagrama de flujo que ilustra un procedimiento de codificación de vídeo de ejemplo de la divulgación. El procedimiento de la Figura 6 puede implementarse mediante el codificador de vídeo 20. El codificador de vídeo 20 puede configurarse para determinar un primer tipo de predicción para un bloque de datos de vídeo en un segmento P (602), y para representar el primer tipo de predicción como un elemento de sintaxis de tipo de predicción de segmento P (604). El codificador de vídeo 20 puede configurarse además para determinar un segundo tipo de predicción para un bloque de datos de vídeo en un segmento B (606), y para representar el segundo tipo de predicción como un elemento de sintaxis de tipo de predicción de segmento B (608). El elemento de sintaxis de tipo de predicción de segmento P y el elemento de sintaxis de tipo de predicción de segmento B especifican un modo de predicción y un tipo de partición. El modo de predicción puede incluir interpredicción e intrapredicción. El tipo de partición puede incluir particiones simétricas y particiones asimétricas.

El codificador de vídeo 20 puede configurarse además para determinar una binarización de segmento P para el elemento de sintaxis de tipo de predicción de segmento P (610), y para determinar una binarización de segmento B para el elemento de sintaxis de tipo de predicción de segmento B, en el que el elemento de sintaxis de tipo de predicción de segmento P y el elemento de sintaxis de tipo de predicción de segmento B se determinan mediante el uso de la misma lógica de binarización (612). El codificador de vídeo 20 puede entonces codificar los datos de vídeo en base a las binarizaciones del elemento de sintaxis de tipo de predicción de segmento P y del elemento de sintaxis de tipo de predicción de segmento B (614).

La codificación de los datos de vídeo puede comprender la binarización del elemento de sintaxis de tipo de predicción de segmento P con las binarizaciones de segmento P determinadas, la binarización del elemento de sintaxis de tipo de predicción de segmento B con las binarizaciones de segmento B determinadas, la aplicación de codificación aritmética binaria adaptativa al contexto (CABAC) al elemento de sintaxis de tipo de predicción de segmento P binarizado y la aplicación de codificación aritmética binaria adaptativa al contexto (CABAC) al elemento de sintaxis de tipo de predicción de segmento B binarizado.

La Figura 7 es un diagrama de flujo que ilustra un procedimiento de decodificación de vídeo de ejemplo de la divulgación. El procedimiento de la Figura 7 puede implementarse mediante el decodificador de vídeo 30. El decodificador de vídeo 30 puede configurarse para recibir un elemento de sintaxis de tipo de predicción de segmento P codificado aritméticamente binario adaptativo al contexto que indica el tipo de predicción para el bloque de datos de vídeo en un segmento P (702), y para recibir un elemento de sintaxis de tipo de predicción de segmento B codificado aritméticamente binario adaptativo al contexto que indica el tipo de predicción para el bloque de datos de vídeo en un segmento B (704). El elemento de sintaxis de tipo de predicción de segmento P y el elemento de sintaxis de tipo de predicción de segmento B especifican un modo de predicción y un tipo de partición. El modo de predicción puede incluir interpredicción e intrapredicción. El tipo de partición puede incluir particiones simétricas y particiones asimétricas.

El decodificador de vídeo 30 puede configurarse además para decodificar el elemento de sintaxis de tipo de predicción de segmento P para producir un elemento de sintaxis de tipo de predicción de segmento P binarizado (706), y para decodificar el elemento de sintaxis de tipo de predicción de segmento B para producir un elemento de sintaxis de tipo de predicción de segmento B binarizado (708). El decodificador de vídeo 30 puede configurarse además para mapear el elemento de sintaxis de tipo de predicción de segmento P binarizado a un tipo de predicción mediante el uso de un mapeo de binarización para un bloque de datos de vídeo en un segmento P (710), y para mapear el elemento de sintaxis de tipo de predicción de segmento B binarizado a un tipo de predicción mediante el uso del mismo mapeo de binarización para un bloque de datos de vídeo en un segmento B (712). El decodificador de vídeo 30 puede luego decodificar los datos de vídeo en base a los tipos de predicción mapeados (714).

La Figura 8 es un diagrama de flujo que ilustra un procedimiento de codificación de vídeo de ejemplo de la divulgación. El procedimiento de la Figura 8 puede implementarse mediante el codificador de vídeo 20. El codificador de vídeo 20 puede configurarse para determinar un tipo de partición para un modo de predicción para un bloque de datos de vídeo (802) y para codificar un contenedor de tipo de partición de un elemento de sintaxis de tipo de predicción para el bloque de datos de vídeo mediante el uso de la codificación aritmética binaria adaptativa al contexto (CABAC) con un único contexto (804). El único contexto es el mismo para cualquier tipo de partición. En un ejemplo, el tipo de partición es una partición asimétrica y el contenedor de tipo de partición indica si la partición asimétrica se particiona verticalmente o se particiona horizontalmente. Por ejemplo, el contenedor de tamaño de partición indica si una primera partición es un cuarto del tamaño del bloque de datos de vídeo o si la primera partición es tres cuartos del tamaño del bloque de datos de vídeo.

El codificador de vídeo 20 puede configurarse además para codificar un contenedor de tamaño de partición del elemento de sintaxis de tipo de predicción para el bloque de datos de vídeo mediante el uso de CABAC en modo derivación (806).

La Figura 9 es un diagrama de flujo que ilustra un procedimiento de decodificación de vídeo de ejemplo de la divulgación. El procedimiento de la Figura 9 puede implementarse mediante el decodificador de vídeo 30. El decodificador de vídeo 30 puede configurarse para recibir un elemento de sintaxis de tipo de predicción para un bloque de datos de vídeo que ha sido codificado mediante el uso de la codificación aritmética binaria adaptativa al contexto (CABAC), incluyendo el elemento de sintaxis de tipo de predicción un contenedor de tipo de partición que representa un tipo de partición y un contenedor de tamaño de partición que representa un tamaño de partición (902). En un ejemplo, el tipo de partición es una partición asimétrica y el contenedor de tipo de partición indica si la partición asimétrica se particiona verticalmente o se particiona horizontalmente. Por ejemplo, el contenedor de tamaño de partición indica si una primera partición es un cuarto del tamaño del bloque de datos de vídeo o si la primera partición es tres cuartos del tamaño del bloque de datos de vídeo.

El decodificador de vídeo 30 puede configurarse además para decodificar el contenedor de tipo de partición del elemento de sintaxis de tipo de predicción mediante el uso de CABAC con un único contexto, en el que el único contexto es el mismo para cualquier tipo de partición (904), y para decodificar el contenedor de tamaño de partición del elemento de sintaxis de tipo de predicción mediante el uso de CABAC en modo derivación (906).

La Figura 10 es un diagrama de flujo que ilustra un procedimiento de codificación de vídeo de ejemplo de la divulgación. El procedimiento de la Figura 10 puede implementarse mediante el codificador de vídeo 20 o el decodificador de vídeo. A los efectos de la Figura 10, el codificador de vídeo 20 y el decodificador de vídeo 30 se denominarán, colectivamente, un codificador de vídeo. De acuerdo con las técnicas de la Figura 10, un codificador de vídeo puede configurarse para codificar un indicador de bloque codificado por croma Cb para un bloque de datos de vídeo mediante el uso de la codificación aritmética binaria adaptativa al contexto (CABAC), en el que la codificación del indicador de bloque codificado por croma Cb comprende usar un conjunto de contexto que incluye uno o más contextos como parte de la CABAC (1002), y para codificar un indicador de bloque codificado por croma Cr mediante el uso de CABAC, en el que la codificación del indicador de bloque codificado por croma Cr comprende usar el mismo conjunto de contexto que el indicador de bloque codificado por croma Cb como parte de la CABAC (1004). En un ejemplo, el conjunto de contexto incluye 5 contextos.

En un ejemplo opcional de la divulgación, un codificador de vídeo puede configurarse además para seleccionar un contexto de uno o más contextos en base a una profundidad de transformación de una unidad de transformación asociada con el bloque de datos de vídeo (1006).

Cuando funciona como un codificador de vídeo, un codificador de vídeo puede configurarse además para señalar el indicador de bloque codificado por croma Cb codificado en un flujo de bits de vídeo codificado, y para señalar el indicador de bloque codificado por croma Cr codificado en el flujo de bits de vídeo codificado. Cuando funciona como un decodificador de vídeo, un codificador de vídeo puede configurarse además para recibir el indicador de bloque codificado por croma Cb codificado en un flujo de bits de vídeo codificado, y para recibir el indicador de bloque codificado por croma Cr codificado en el flujo de bits de vídeo codificado.

En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, microprograma o cualquiera de sus combinaciones. Si se implementan en software, las funciones pueden almacenarse o transmitirse, como una o más instrucciones o código, en un medio legible por ordenador y ejecutarse por una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible, tal como los medios de almacenamiento de datos o los medios de comunicación, que incluyen cualquier medio que facilite transferir un programa informático de un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador generalmente pueden corresponder a (1) medios de almacenamiento legibles por ordenador tangibles 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 se pueda acceder por uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en esta divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

A manera de ejemplo, y no de limitación, tales medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EPROM, EEPROM, CD-ROM u otro almacenamiento en disco óptico, almacenamiento en disco magnético u otros dispositivos de almacenamiento magnéticos, memoria flash o cualquier otro medio que pueda utilizarse para almacenar el código del programa deseado en forma de instrucciones o estructuras de datos y al que puede accederse por un ordenador. Además, cualquier conexión se denomina apropiadamente como un medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, servidor u otra fuente remota mediante el uso de un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de suscriptor digital (DSL) o tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas se incluyen en la definición de medio. Sin embargo, debe entenderse que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen las conexiones, ondas portadoras, señales u otros medios transitorios, sino que se dirigen a medios de almacenamiento tangibles no transitorios. Disco magnético y disco óptico, como se usa en la presente memoria, incluye disco compacto (CD), disco láser, disco óptico, disco versátil digital (DVD), disquete y disco Blu-ray, donde los discos suelen reproducir datos magnéticamente, mientras que los discos reproducen datos ópticamente con láser. Las combinaciones de los medios anteriores también pueden incluirse dentro del ámbito de los medios legibles por ordenador.

Las instrucciones pueden ejecutarse por uno o más procesadores, como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados para aplicaciones específicas (ASIC), matrices lógicas programables en campo (FPGA) u otro circuito integrado lógico o discreto equivalente. En consecuencia, el término "procesador", como se usa en la presente memoria puede referirse a cualquiera de las estructuras anteriores o cualquier otra estructura adecuada para la implementación de las técnicas descritas en la presente memoria. Además, en algunos aspectos, la funcionalidad descrita en la presente memoria puede proporcionarse dentro de módulos de hardware y/o software dedicados configurados para la codificación y decodificación, o incorporarse en un códec combinado. Asimismo, las técnicas se podrían implementar 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 IC (por ejemplo, un conjunto de chips).

En esta divulgación se describen diversos componentes, módulos o unidades para enfatizar los aspectos funcionales de los dispositivos configurados para realizar las técnicas divulgadas, pero no necesariamente requieren la realización por diferentes unidades de hardware. Más bien, como se describió anteriormente, diversas unidades pueden combinarse en una unidad de hardware de códec o proporcionarse por una colección de unidades de hardware interoperativas, que incluyen uno o más procesadores como se describió anteriormente, junto con software y/o microprograma adecuados.

Se describen diversos ejemplos. Estos y otros ejemplos están dentro del ámbito de las siguientes reivindicaciones.

10

15

20

25

30

35

40

45

50

55

60

65

REIVINDICACIONES

1. Un procedimiento de decodificación de datos de vídeo, el procedimiento que comprende:

5 decodificar un elemento de sintaxis de tipo de predicción de segmento P codificado mediante la codificación aritmética binaria adaptativa al contexto, CABAC, para producir un elemento de sintaxis de tipo de predicción de segmento P binarizado;
 decodificar un elemento de sintaxis de tipo de predicción de segmento B codificado mediante CABAC para producir un elemento de sintaxis de tipo de predicción de segmento B binarizado;
 10 decodificar un elemento de sintaxis de tipo de predicción de segmento I codificado mediante CABAC para producir un elemento de sintaxis de tipo de predicción de segmento I binarizado;
 mapear el elemento de sintaxis de tipo de predicción de segmento P binarizado a un primer tipo de predicción mediante el uso de un primer mapeo de binarización para un bloque de datos de vídeo en un segmento P, en el que el primer mapeo de binarización mapea una primera pluralidad de cadenas binarizadas a una primera pluralidad de tipos de predicción, y la primera pluralidad de tipos de predicción incluye el primer tipo de predicción;
 15 mapear el elemento de sintaxis de tipo de predicción de segmento B binarizado a un segundo tipo de predicción mediante el uso del primer mapeo de binarización para un bloque de datos de vídeo en un segmento B, en el que la primera pluralidad de tipos de predicción incluye el segundo tipo de predicción;
 20 mapear el elemento de sintaxis de tipo de predicción de segmento I binarizado a un tercer tipo de predicción mediante el uso de un segundo mapeo de binarización para un bloque de datos de vídeo en un segmento I, en el que el segundo mapeo de binarización mapea una segunda pluralidad de cadenas binarizadas a una segunda pluralidad de tipos de predicción, el segundo mapeo de binarización es diferente del primer mapeo de binarización, la segunda pluralidad de tipos de predicción es diferente de la primera pluralidad de tipos de predicción, y la segunda pluralidad de tipos de predicción incluye el tercer tipo de predicción,
 25 en el que cada uno del primer tipo de predicción, el segundo tipo de predicción y el tercer tipo de predicción especifica un modo de predicción respectivo y un tipo de partición respectivo; y
 decodificar los datos de vídeo en base al primer tipo de predicción, el segundo tipo de predicción y el tercer tipo de predicción mapeados.

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

recibir el elemento de sintaxis de tipo de predicción de segmento P codificado mediante CABAC que indica el primer tipo de predicción para el bloque de datos de vídeo en el segmento P;
 35 recibir el elemento de sintaxis de tipo de predicción de segmento B codificado mediante CABAC que indica el segundo tipo de predicción para el bloque de datos de vídeo en el segmento B; y
 recibir el elemento de sintaxis de tipo de predicción de segmento I codificado mediante CABAC que indica el tercer tipo de predicción para el bloque de datos de vídeo en el segmento I.

3. El procedimiento de las reivindicaciones 1-2, en el que el modo de predicciones incluye uno de interpredicción e intrapredicción.

4. El procedimiento de las reivindicaciones 1-3, en el que el tipo de partición incluye una de particiones simétricas y particiones asimétricas.

5. El procedimiento de las reivindicaciones 1-4, en el que un número de tipos de predicción en la primera pluralidad de tipos de predicción es diferente de un número de tipos de predicción en la segunda pluralidad de tipos de predicción.

6. Un aparato configurado para decodificar los datos de vídeo, el aparato que comprende:

medios para decodificar un elemento de sintaxis de tipo de predicción de segmento P codificado mediante codificación aritmética binaria adaptativa al contexto, CABAC, para producir un elemento de sintaxis de tipo de predicción de segmento P binarizado;
 55 medios para decodificar un elemento de sintaxis de tipo de predicción de segmento B codificado mediante CABAC para producir un elemento de sintaxis de tipo de predicción de segmento B binarizado;
 medios para decodificar un elemento de sintaxis de tipo de predicción de segmento I codificado mediante CABAC para producir un elemento de sintaxis de tipo de predicción de segmento I binarizado;
 medios para mapear el elemento de sintaxis de tipo de predicción de segmento P binarizado a un primer tipo de predicción mediante el uso de un primer mapeo de binarización para un bloque de datos de vídeo en un segmento P, en el que el primer mapeo de binarización mapea una primera pluralidad de cadenas binarizadas a una primera pluralidad de tipos de predicción, y la primera pluralidad de tipos de predicción incluye el primer tipo de predicción;
 60 medios para mapear el elemento de sintaxis de tipo de predicción de segmento B binarizado a un segundo tipo de predicción mediante el uso del primer mapeo de binarización para un bloque de datos de vídeo en un segmento B, en el que la primera pluralidad de tipos de predicción incluye el segundo tipo de predicción;

medios para mapear el elemento de sintaxis de tipo de predicción de segmento I binarizado a un tercer tipo de predicción mediante el uso de un segundo mapeo de binarización para un bloque de datos de vídeo en un segmento I, en el que el segundo mapeo de binarización mapea una segunda pluralidad de cadenas binarizadas a una segunda pluralidad de tipos de predicción, el segundo mapeo de binarización es diferente del primer mapeo de binarización, la segunda pluralidad de tipos de predicción es diferente de la primera pluralidad de tipos de predicción, y la segunda pluralidad de tipos de predicción incluye el tercer tipo de predicción, en el que cada uno del primer tipo de predicción, el segundo tipo de predicción y el tercer tipo de predicción especifica un modo de predicción respectivo y un tipo de partición respectivo; y medios para decodificar los datos de vídeo en base al primer tipo de predicción, el segundo tipo de predicción y el tercer tipo de predicción mapeados.

7. El aparato de la reivindicación 6, que comprende, además:

medios para recibir el elemento de sintaxis de tipo de predicción de segmento P codificado mediante CABAC que indica el primer tipo de predicción para el bloque de datos de vídeo de segmento P; medios para recibir el elemento de sintaxis de tipo de predicción de segmento B codificado mediante CABAC que indica el segundo tipo de predicción para el bloque de datos de vídeo en el segmento B; y medios para recibir el elemento de sintaxis de tipo de predicción de segmento I codificado mediante CABAC que indica el tercer tipo de predicción para el bloque de datos de vídeo y en el segmento I.

8. El aparato de las reivindicaciones 6-7, en el que el modo de predicción incluye uno de interpredicción e intrapredicción.

9. El aparato de las reivindicaciones 6-8, en el que el tipo de partición incluye una de particiones simétricas y particiones asimétricas.

10. El aparato de las reivindicaciones 6-9, en el que un número de tipos de predicción en la primera pluralidad de tipos de predicción es diferente de un número de tipos de predicción en la segunda pluralidad de tipos de predicción.

11. Un medio de almacenamiento legible por ordenador no transitorio que almacena instrucciones que, cuando se ejecutan, provocan que uno o más procesadores configurados para decodificar datos de vídeo para:

decodificar un elemento de sintaxis de tipo de predicción de segmento P codificado mediante la codificación aritmética binaria adaptativa al contexto, CABAC, para producir un elemento de sintaxis de tipo de predicción de segmento P binarizado; decodificar un elemento de sintaxis de tipo de predicción de segmento B codificado mediante CABAC para producir un elemento de sintaxis de tipo de predicción de segmento B binarizado; decodificar un elemento de sintaxis de tipo de predicción de segmento I codificado mediante CABAC para producir un elemento de sintaxis de tipo de predicción de segmento I binarizado; mapear el elemento de sintaxis de tipo de predicción de segmento P binarizado a un primer tipo de predicción mediante el uso de un primer mapeo de binarización para un bloque de datos de vídeo en un segmento P, en el que el primer mapeo de binarización mapea una primera pluralidad de cadenas binarizadas a una primera pluralidad de tipos de predicción, y la primera pluralidad de tipos de predicción incluye el primer tipo de predicción; mapear el elemento de sintaxis de tipo de predicción de segmento B binarizado a un segundo tipo de predicción mediante el uso del primer mapeo de binarización para un bloque de datos de vídeo en un segmento B, en el que la primera pluralidad de tipos de predicción incluye el segundo tipo de predicción; mapear el elemento de sintaxis de tipo de predicción de segmento I binarizado a un tercer tipo de predicción mediante el uso de un segundo mapeo de binarización para un bloque de datos de vídeo en un segmento I, en el que el segundo mapeo de binarización mapea una segunda pluralidad de cadenas binarizadas a una segunda pluralidad de tipos de predicción, el segundo mapeo de binarización es diferente del primer mapeo de binarización, la segunda pluralidad de tipos de predicción es diferente de la primera pluralidad de tipos de predicción, y la segunda pluralidad de tipos de predicción incluye el tercer tipo de predicción, en el que cada uno del primer tipo de predicción, el segundo tipo de predicción y el tercer tipo de predicción especifica un modo de predicción respectivo y un tipo de partición respectivo; y decodificar los datos de vídeo en base al primer tipo de predicción, el segundo tipo de predicción y el tercer tipo de predicción mapeados.

12. El medio de almacenamiento legible por ordenador no transitorio de la reivindicación 11, en el que las instrucciones hacen además que uno o más procesadores:

reciba el elemento de sintaxis de tipo de predicción de segmento P codificado mediante CABAC que indica el primer tipo de predicción para el bloque de datos de vídeo en el segmento P; reciba el elemento de sintaxis de tipo de predicción de segmento B codificado mediante CABAC que indica el

segundo tipo de predicción para el bloque de datos de vídeo en el segmento B; y
 reciba el elemento de sintaxis de tipo de predicción de segmento I codificado mediante CABAC que indica el
 tercer tipo de predicción para el bloque de datos de vídeo en el segmento I.

- 5 13. El medio de almacenamiento legible por ordenador no transitorio de las reivindicaciones 11-12, en el que el modo
 de predicción incluye uno de interpredicción e intrapredicción, y/o en el que el tipo de partición incluye uno de
 particiones simétricas y particiones asimétricas.
- 10 14. El medio de almacenamiento legible por ordenador no transitorio de las reivindicaciones 11-13, en el que un
 número de tipos de predicción en la primera pluralidad de tipos de predicción es diferente de un número de tipos
 de predicción en la segunda pluralidad de tipos de predicción.

15

20

25

30

35

40

45

50

55

60

65

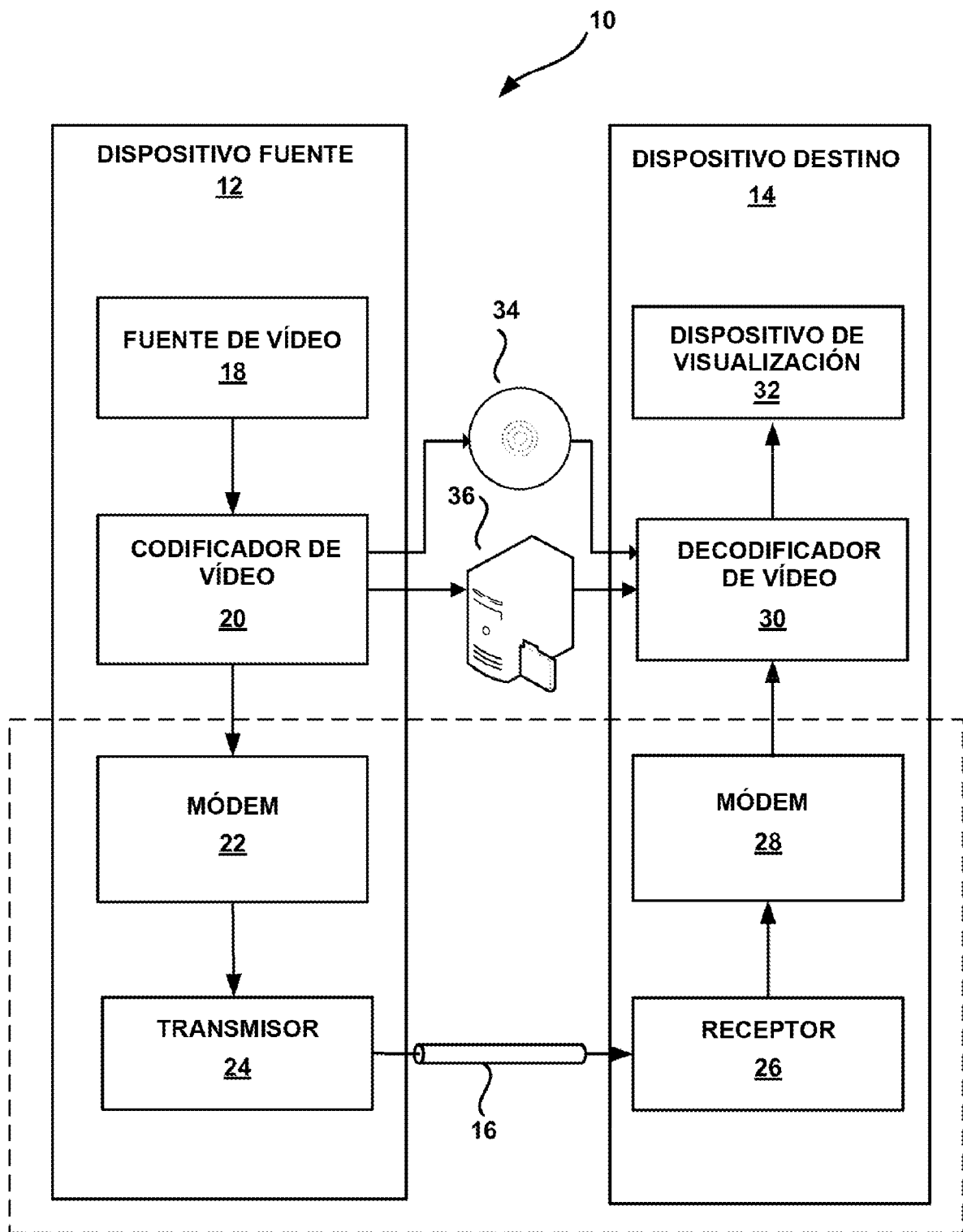


FIGURA 1

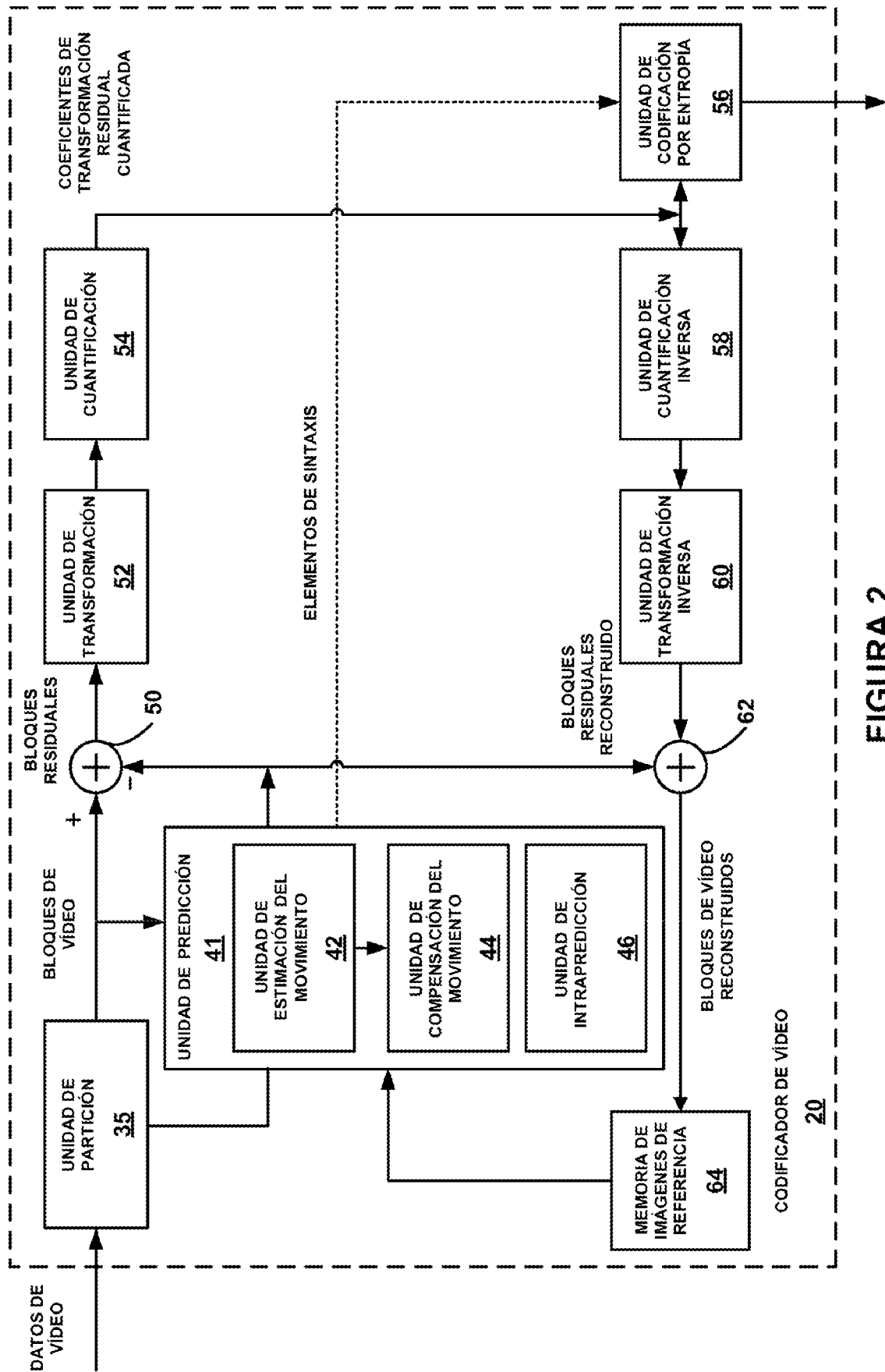


FIGURA 2

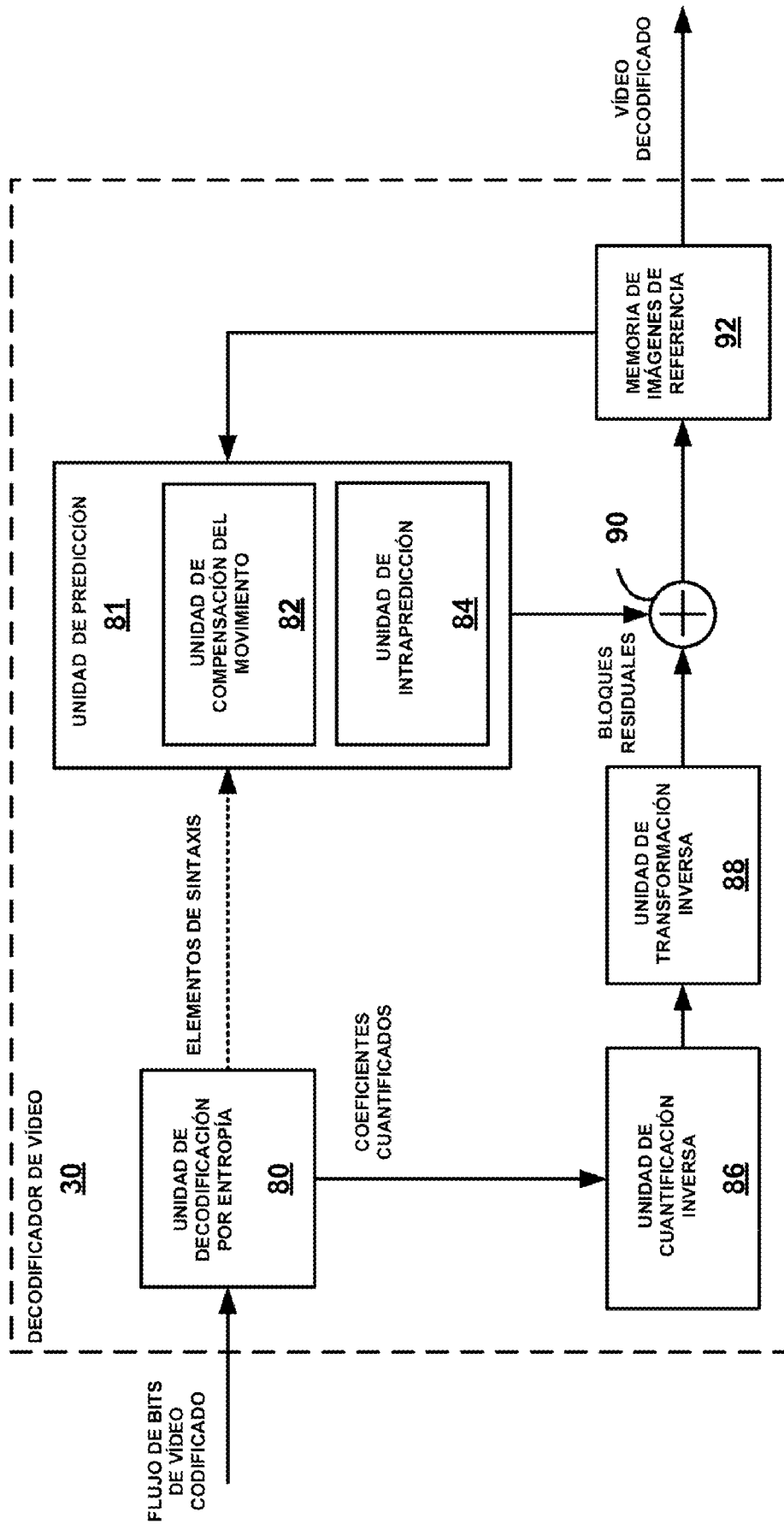


FIGURA 3

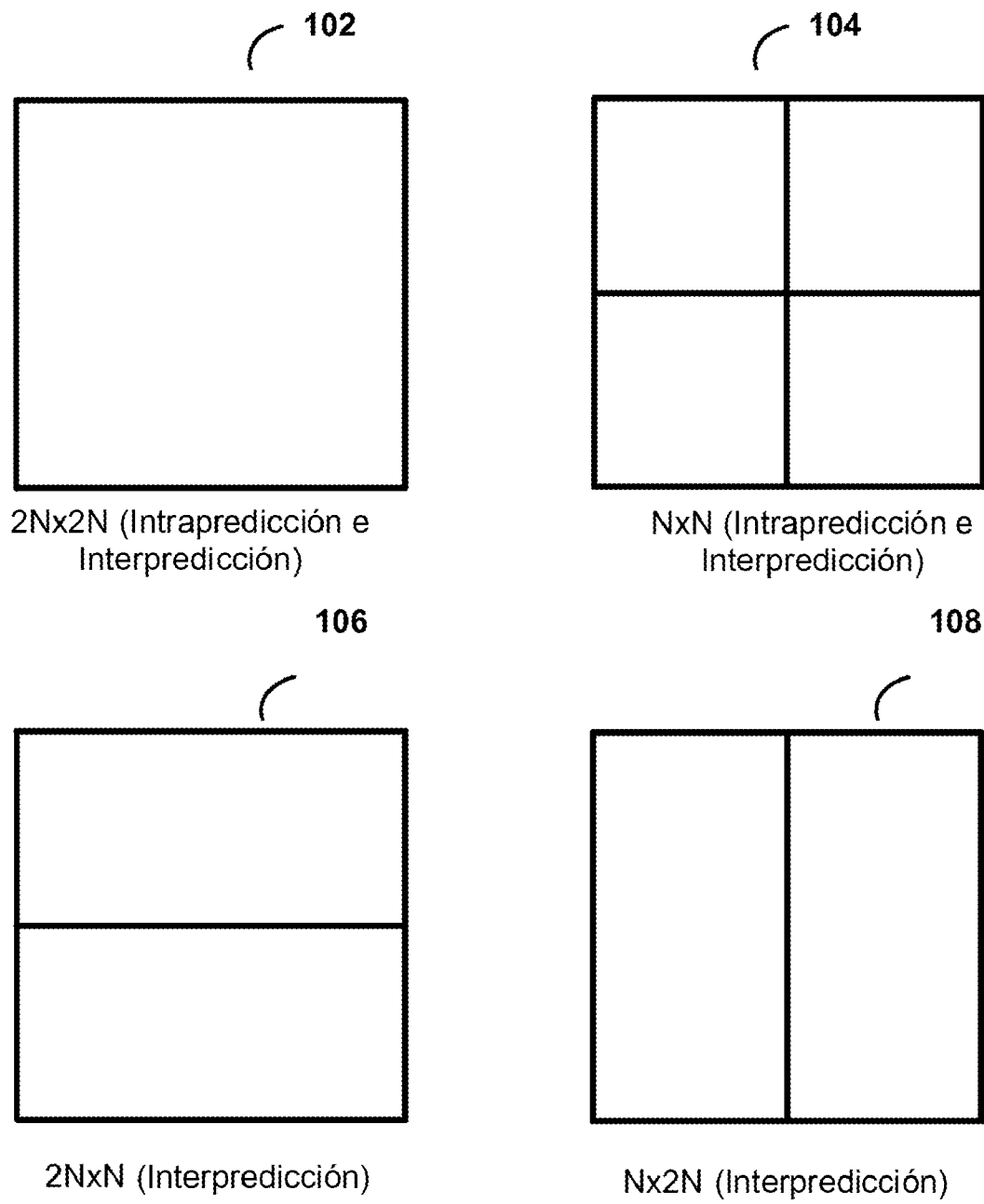


FIGURA 4

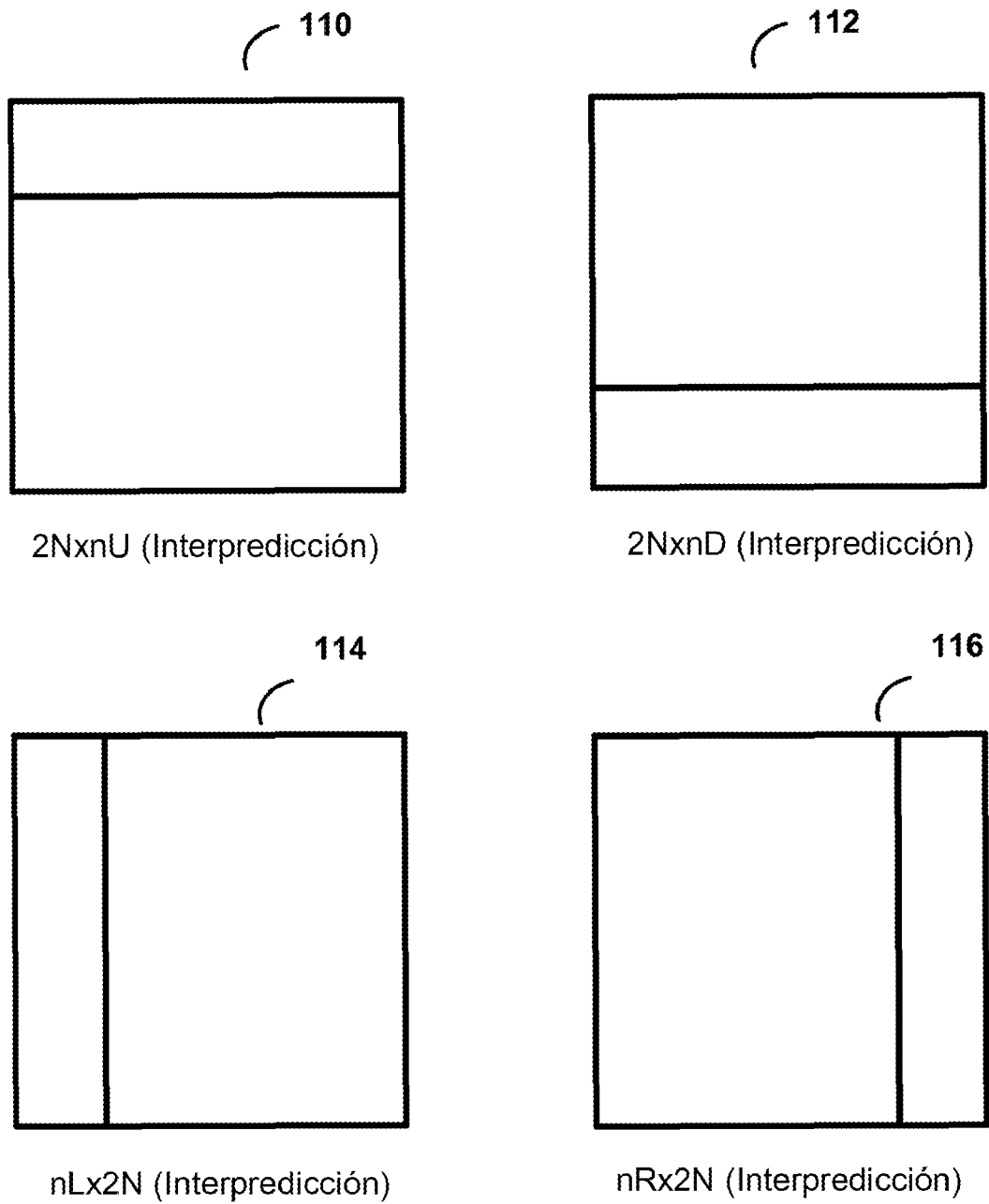
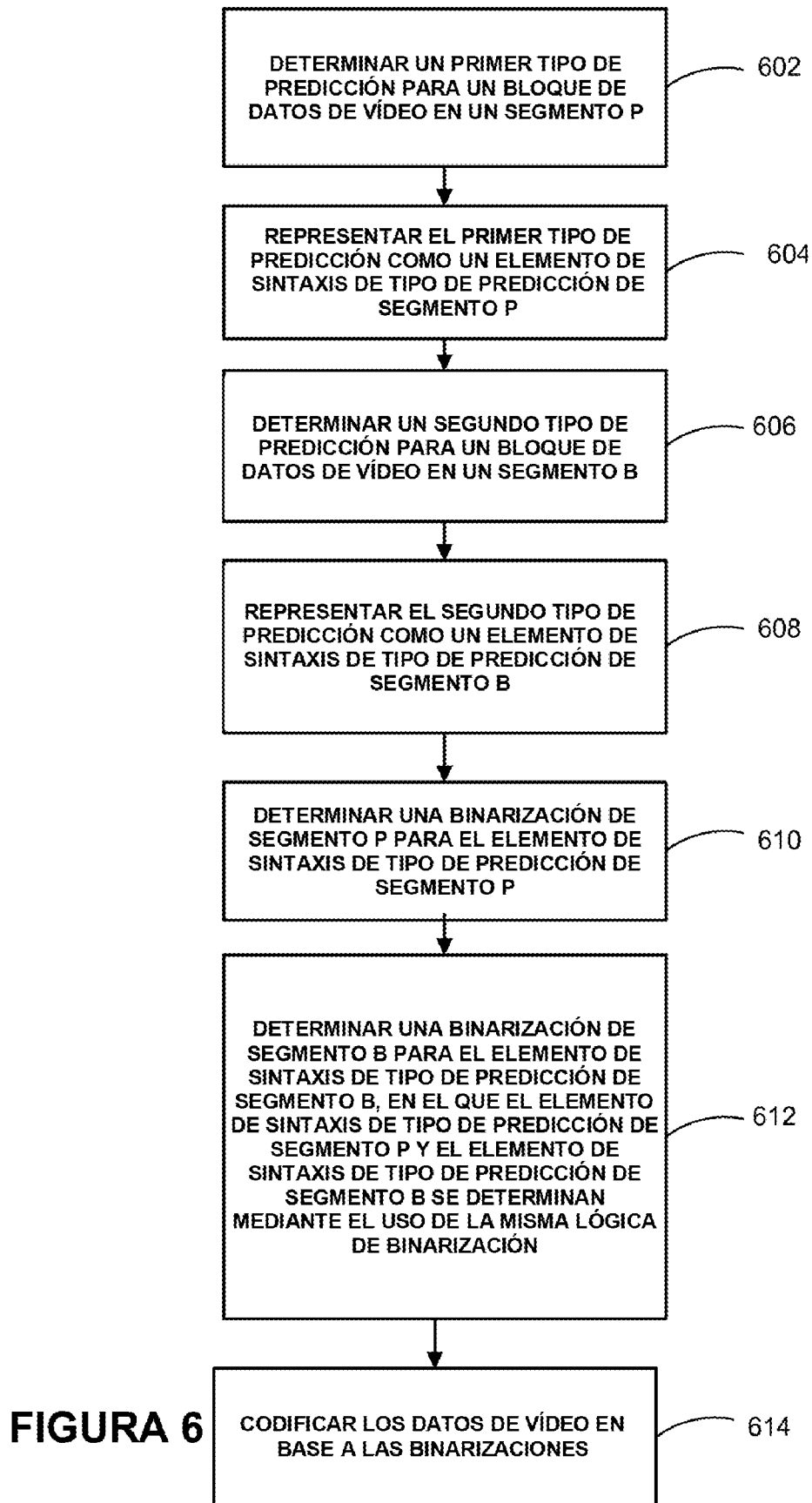
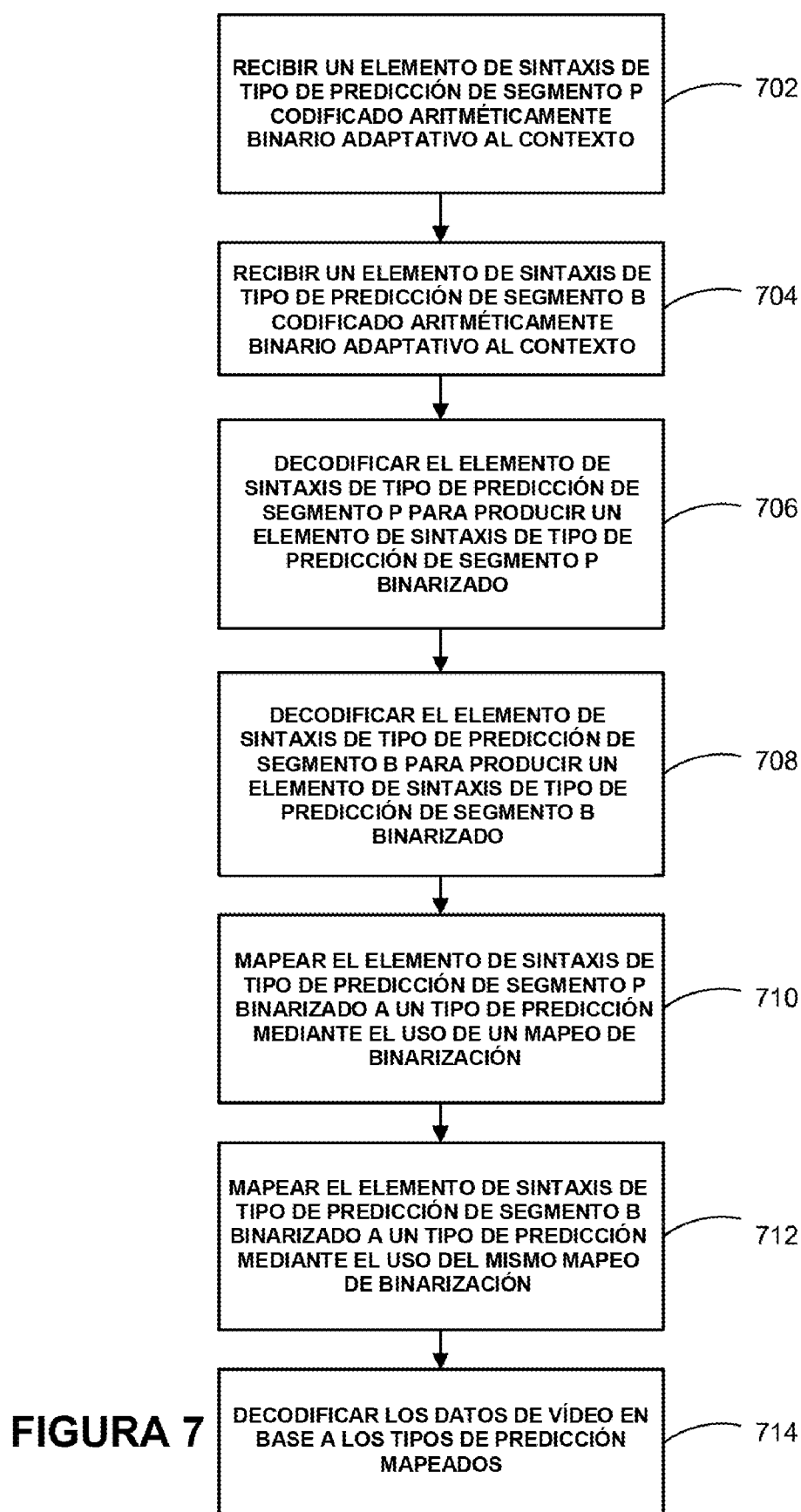


FIGURA 5





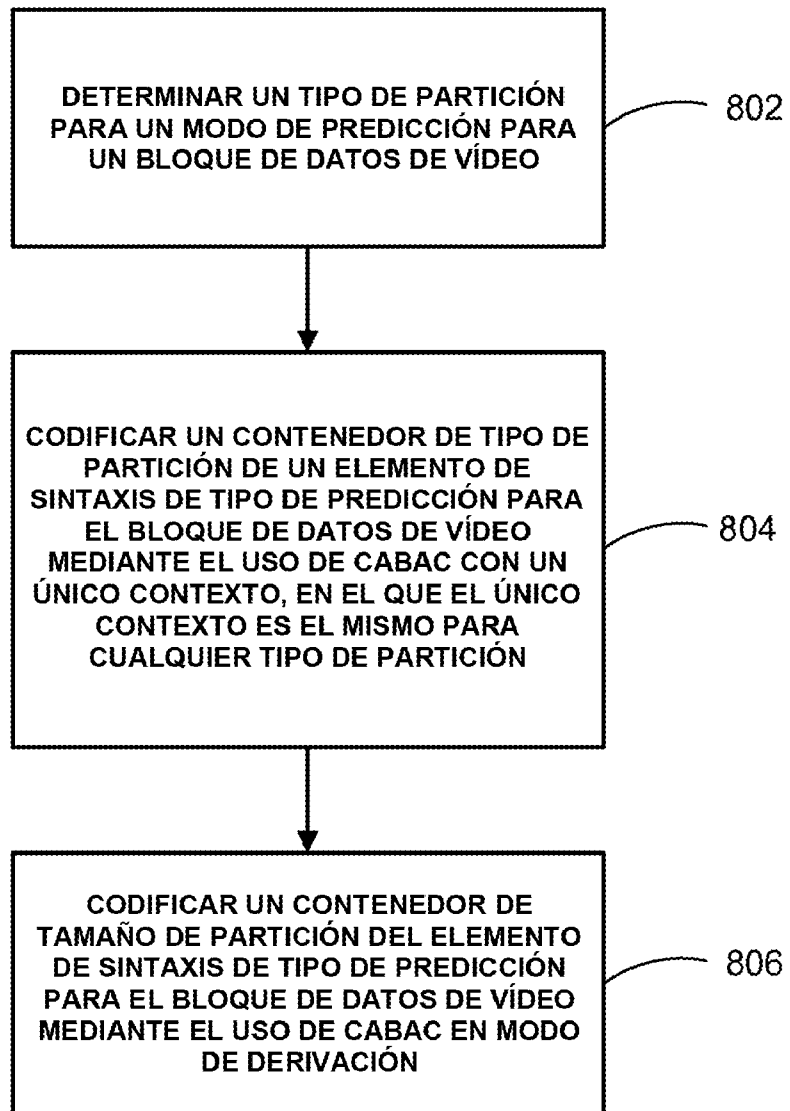


FIGURA 8

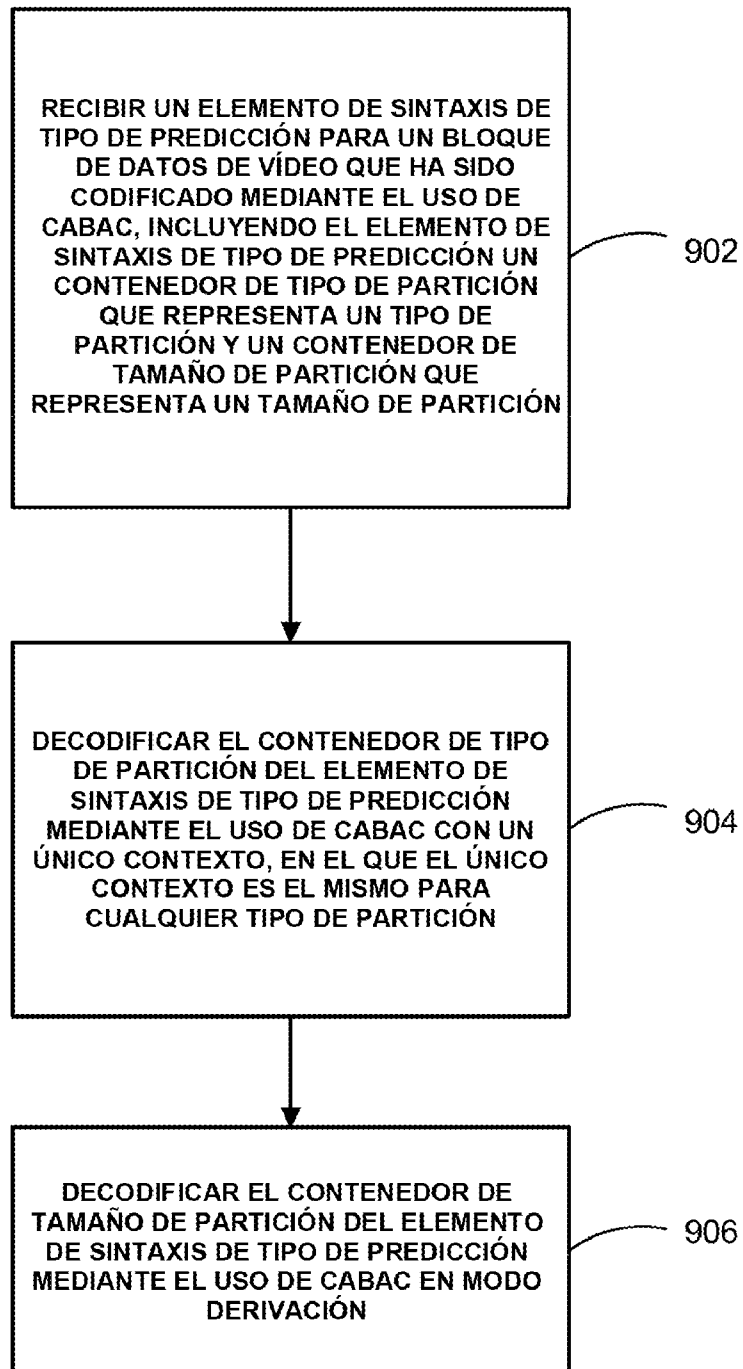


FIGURA 9

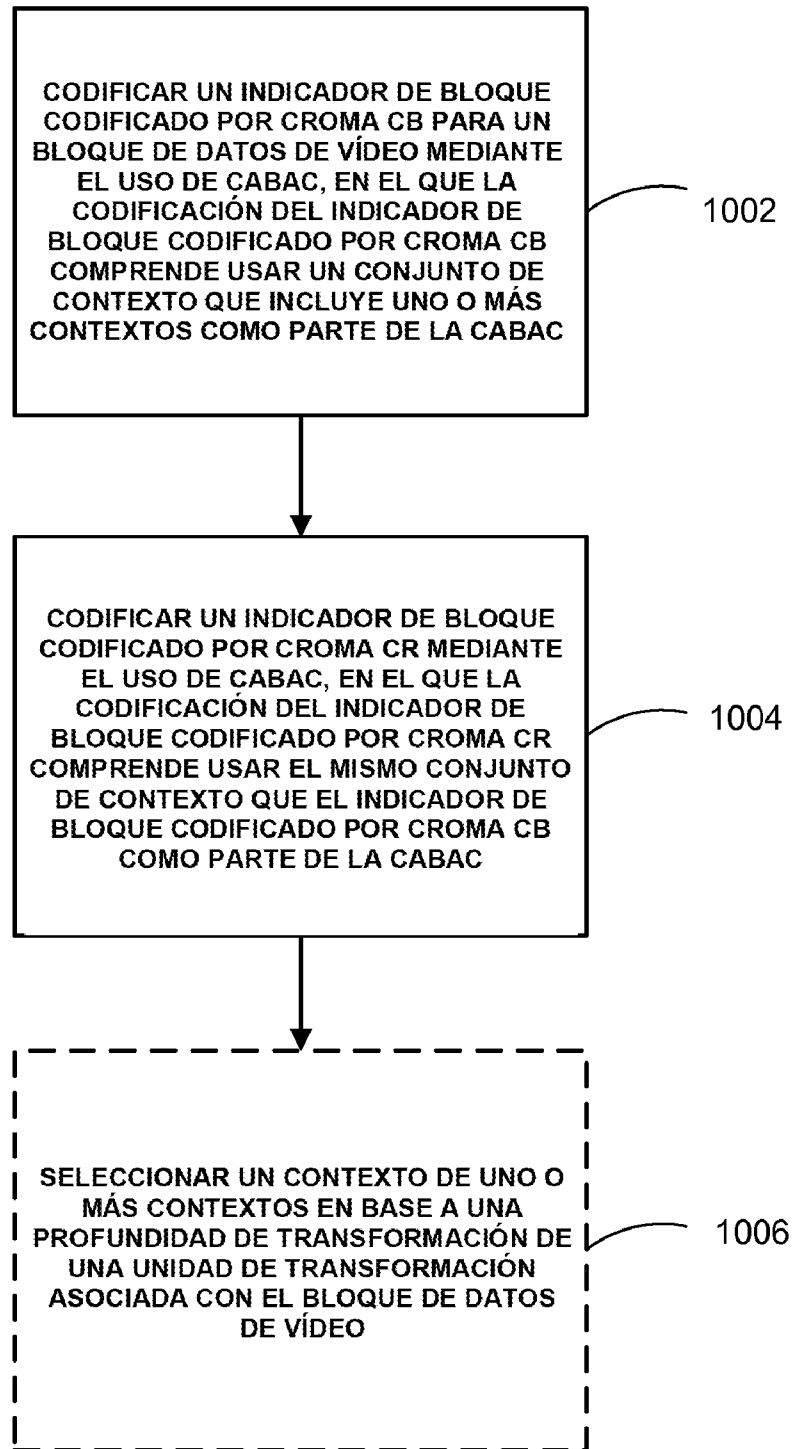


FIGURA 10