

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 550 803**

51 Int. Cl.:

H04N 19/50 (2014.01)

H04N 19/176 (2014.01)

H04N 19/91 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **05.10.2012 E 12780349 (2)**

97 Fecha y número de publicación de la concesión europea: **26.08.2015 EP 2777160**

54 Título: **Reducción del número de contextos para codificación aritmética binaria adaptativa al contexto**

30 Prioridad:

08.11.2011 US 201161557325 P

20.11.2011 US 201161561911 P

04.10.2012 US 201213645330

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

12.11.2015

73 Titular/es:

QUALCOMM INCORPORATED (100.0%)

5775 Morehouse Drive

San Diego, CA 92121-1714, US

72 Inventor/es:

CHIEN, WEI-JUNG;

SOLE ROJALS, JOEL y

KARCZEWICZ, MARTA

74 Agente/Representante:

CARPINTERO LÓPEZ, Mario

ES 2 550 803 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

Reducción del número de contextos para codificación aritmética binaria adaptativa al contexto

Campo técnico

5 La presente divulgación se refiere a codificación de vídeo, y en particular a codificación aritmética binaria (CABAC) adaptativa según el contexto utilizado en la codificación de vídeo.

Antecedentes

10 Las capacidades de vídeo digital se pueden incorporar en una amplia gama de dispositivos, incluyendo televisores digitales, sistemas de radiodifusión directa digitales, sistemas de transmisión inalámbricos, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, ordenadores de tableta, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digitales, reproductores multimedia digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos celulares o satelitales de radio, los llamados "teléfonos inteligentes", dispositivos de teleconferencia de vídeo, dispositivos de vídeo de transmisión en vivo, y similares. Los dispositivos de vídeo digitales implementan técnicas de compresión de vídeo, tales como las descritas en los estándares definidos por MPEG-2, MPEG-4, UIT-T H.263, UIT-T H.264/MPEG-4, Parte 10, codificación de vídeo avanzada (AVC), estándar de codificación de vídeo de alta eficiencia (HEVC) actualmente en fase de desarrollo, y extensiones de esas normas. Los dispositivos de vídeo pueden transmitir, recibir, codificar, decodificar, y/o almacenar información de vídeo digital de manera más eficiente mediante la implementación de este tipo de técnicas de compresión de vídeo.

20 Las técnicas de compresión de vídeo realizan predicción espacial (dentro de las imágenes) y/o una predicción temporal (entre imágenes) para reducir o eliminar la redundancia inherente a las secuencias de vídeo. Para la codificación de vídeo basado en bloques, una porción de vídeo (es decir, una imagen de vídeo o una porción de una imagen de vídeo) puede dividirse en bloques de vídeo, que también pueden denominarse como bloques de árbol, unidades de codificación (CUS) y/o nodos de codificación. Los bloques de vídeos en una porción intra-codificada (I) de una imagen se codifican utilizando predicción espacial con respecto a las muestras de referencia en bloques vecinos de la misma imagen. Los bloques de vídeos en una porción inter-codificada (P o B) de una imagen pueden usar predicción espacial con respecto a las muestras de referencia en bloques vecinos de la misma imagen o predicción temporal con respecto a las muestras de referencia en otras imágenes de referencia. Las imágenes pueden indicarse como marcos, imágenes de referencia y pueden indicarse como marcos de referencia.

30 La predicción espacial o temporal resulta en un bloque de predicción para un bloque a codificar. Los datos residuales representan diferencias de píxeles entre el bloque original a codificar y el bloque de predicción. Un bloque inter-codificado se codifica de acuerdo con un vector de movimiento que apunta a un bloque de muestras de referencia que forman el bloque de predicción, y los datos residuales que indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intra-codificado se codifica de acuerdo con un modo de intra-codificación y los datos residuales. Para mayor compresión, los datos residuales pueden ser transformados desde el dominio de píxeles a un dominio de transformación, dando como resultado coeficientes de transformación residuales, que luego pueden ser cuantificados. Los coeficientes de transformación cuantificados, inicialmente dispuestos en una matriz de dos dimensiones, pueden ser escaneados para producir un vector unidimensional de coeficientes de transformación, y la codificación de entropía se puede aplicar para lograr aún más compresión. Los siguientes artículos se refieren a reducción de sitio de contexto para CABAC.

40 "BoG report on context reduction for CABAC", 6 JCT-VC Meeting; 97. MPEG Meeting, 14-22 de julio de 2011, Turín;
 "Modified binarization and coding of MVD for PIPE/CABAC" de Nguyen et al, en 6. JCT-VC Meeting; 97. MPEG Meeting; 14-22 de Julio de 2011, Turín; y
 45 "CE11: Context size reduction for the significance map" de Hisao Sasai et al, en 96. MPEG Meeting; 21-25 de marzo de 2011, Ginebra.

Sumario

50 La invención se define en las reivindicaciones a las que se hace ahora referencia. En general, esta divulgación describe técnicas para la codificación aritmética binaria (CABAC) adaptativa según contexto en un proceso 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, ejemplos no limitativos de los cuales incluyen *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 pueden reducir hasta 56 contextos con insignificantes cambios de eficiencia de codificación. Las reducciones de contexto propuestas para los elementos de sintaxis pueden ser utilizadas en solitario o en cualquier fusión.

55 La codificación de vídeo puede incluir la determinación de un primer tipo de predicción para un bloque de datos de vídeo en una porción P, que representa el primer tipo de predicción como un elemento de sintaxis de tipo de predicción de porción P, la determinación de un segundo tipo de predicción para un bloque de datos de vídeo en una

porción B, la representación del segundo tipo de predicción como un elemento de sintaxis de tipo de predicción de porción B, la determinación de una binarización de porción P para el elemento de sintaxis de tipo de predicción de porción P, la determinación una binarización de porción B para el elemento de sintaxis de tipo de predicción de porción B, en el que el elemento de sintaxis de tipo de predicción de porción P y el elemento de sintaxis de tipo de predicción de porción B se determinan usando la misma lógica de binarización, y la codificación de los datos de vídeo basados en los binarizaciones del elemento de sintaxis del tipo de predicción de porción P y el elemento de sintaxis de tipo de predicción de porción B.

La decodificación de vídeo puede incluir la asignación de un elemento de sintaxis de tipo de predicción de porción P binarizada a un tipo de predicción usando una asignación de binarización para un bloque de datos de vídeo en una porción P, la asignación de un elemento de sintaxis tipo de predicción de porción B binarizada a un tipo de predicción utilizando la misma asignación de binarización para un bloque de datos de vídeo en una porción B, y la decodificación de los datos de vídeo en base a los tipos de predicción asignados.

La codificación de datos de vídeo puede comprender la determinación de un tipo de partición para un modo de predicción para un bloque de datos de vídeo, que codifica una partición de tipo bin de un elemento de sintaxis de tipo de predicción para un bloque de datos de vídeo utilizando CABAC con un solo contexto, en el que el contexto individual es el mismo para cualquier tipo de partición, y que codifica una partición de tamaño bin del elemento de sintaxis de tipo de predicción para el bloque de datos de vídeo utilizando CABAC en el modo de derivación.

La decodificación de datos de vídeo puede comprender la recepción de un elemento de sintaxis de tipo de predicción para un bloque de datos de vídeo que ha sido codificado usando CABAC, incluyendo el elemento de sintaxis de tipo de predicción un tipo de partición bin que representa un tipo de partición y un tamaño de partición bin que representa un tamaño de partición, decodificando el tipo de partición bin del elemento de sintaxis del tipo de predicción usando el contexto de codificación aritmética binaria adaptativa con un solo contexto, en el que el contexto individual es el mismo para cualquier tipo de partición, y decodificando el tamaño de partición bin del elemento de sintaxis de tipo de predicción utilizando CABAC en modo de derivación.

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

Breve descripción de los dibujos

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

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

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

La figura 5 es un dibujo conceptual que muestra los tipos de partición asimétricos.

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 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

Esta divulgación describe técnicas de codificación de datos, tal como datos de vídeo. En particular, la divulgación describe técnicas que pueden promover la codificación eficiente de datos de vídeo utilizando procesos de codificación de entropía de contexto adaptativo. Más específicamente, esta divulgación propone una reducción en el número de contextos CABAC utilizados para elementos de sintaxis de codificación, 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 insignificantes cambios de eficiencia de codificación. Esta divulgación describe la codificación de vídeo para fines de ilustración. Sin embargo, las técnicas descritas en esta divulgación también pueden ser aplicables a otros tipos de codificación de datos.

La figura 1 es un diagrama de bloques que ilustra un ejemplo de sistema 10 de codificación y decodificación de vídeo que puede configurarse para utilizar técnicas para codificación aritmética binaria (CABAC) adaptativa según el contexto de acuerdo con ejemplos de esta divulgación. Como se muestra en la figura 1, el sistema 10 incluye un dispositivo 12 fuente que transmite vídeo codificado a un dispositivo 14 de destino a través del canal 16 de

comunicación. Los datos de vídeo codificados también se pueden almacenar en un medio 34 de almacenamiento o servidor 36 de archivos y se puede acceder mediante el dispositivo 14 de destino 14 si se desea. Cuando se almacena en un medio de almacenamiento o servidor de archivos, un codificador 20 de vídeo puede proporcionar datos de vídeo codificados a otro dispositivo, tal como una interfaz de red, un dispositivo de instalación de grabación o registro de discos compactos (CD), Blu-ray o discos de vídeo digital (DVD), u otros dispositivos, para almacenar los datos de vídeo codificados en el medio de almacenamiento. Del mismo modo, un dispositivo separado de decodificador 30 de vídeo, tal como un lector de interfaz de red, CD o DVD, o similar, puede recuperar los datos codificados de vídeo desde un medio de almacenamiento y proporcionar los datos recuperados al decodificador 30 de vídeo.

Dispositivo 12 fuente y dispositivo 14 destino pueden comprender cualquiera de una amplia variedad de dispositivos, incluyendo ordenadores de escritorio, ordenadores portátiles (por ejemplo, portátil), ordenadores de tableta, codificadores, aparatos telefónicos, tales como los llamados teléfonos inteligentes, televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, o similares. En muchos casos, dichos dispositivos pueden estar equipados para la comunicación inalámbrica. Por lo tanto, el canal 16 de comunicación puede comprender un canal inalámbrico, un canal por cable, o una fusión de canales inalámbricos y cableados adecuados para la transmisión de datos de vídeo codificados. Del mismo modo, un servidor 36 de archivos puede ser accedido por el dispositivo 14 de destino a través de cualquier conexión de datos estándar, incluyendo una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión Wi-Fi), una conexión por cable (por ejemplo, DSL, cable módem, etc.), o una fusión de ambos, que es adecuado para el acceso a los datos de vídeo codificados almacenados en un servidor de archivos.

Las técnicas para CABAC, de acuerdo con los ejemplos de esta divulgación, se pueden aplicar a la codificación de vídeo en apoyo de cualquiera de una variedad de aplicaciones multimedia, como emisiones de televisión en el aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisión de transmisiones de vídeo, por ejemplo, a través de Internet, codificación de vídeo digital para el 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 puede configurarse para soportar transmisión de vídeo en un solo sentido o en dos sentidos para soportar aplicaciones tales como transmisión de vídeo, reproducción de vídeo, retransmisión de vídeo y/o videotelefonía.

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

El vídeo capturado, pre-capturado, o generado por ordenador puede ser codificado por el codificador de vídeo 20. La información de vídeo codificada puede ser modulada por módem 22 de acuerdo con un estándar de comunicación, tal como un protocolo de comunicación inalámbrica, y se transmite al dispositivo 14 de destino 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 la transmisión de datos, incluyendo amplificadores, filtros, y una o más antenas.

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

El servidor 36 de archivos puede ser cualquier tipo de servidor capaz de almacenar vídeo codificado y transmitir ese vídeo codificado al dispositivo 14 de destino. Servidores de archivos de ejemplo incluyen un servidor web (por ejemplo, para un sitio web), un servidor FTP, un almacenamiento conectado a la red (NAS), una unidad de disco local, o cualquier otro tipo de dispositivo capaz de almacenar los datos de vídeo codificados y transmitirlos a un dispositivo de destino. La transmisión de datos de vídeo codificados de servidor de archivos 36 puede ser una transmisión en directo, una transmisión de descarga, o una fusión de ambos. El servidor de archivos 36 se puede acceder mediante el dispositivo 14 de destino a través de cualquier conexión de datos estándar, incluyendo una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión Wi-Fi), una conexión por cable (por ejemplo, DSL, cable módem, Ethernet, USB, etc.), o una fusión de ambos que es adecuada para el acceso a los datos de vídeo codificados almacenados en un servidor de archivos.

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

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

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

El codificador 20 de vídeo y el decodificador 30 de vídeo pueden funcionar de acuerdo con un estándar de compresión de vídeo, tal como el estándar de alta eficiencia de codificación de vídeo (HEVC) actualmente en fase de desarrollo por el Equipo de Colaboración Conjunta de Codificación de Vídeo (JCT-VC) del Grupo de Expertos de Codificación de Vídeo UIT-T (VCEG) y el Grupo de Expertos de Imagen en Movimiento ISO/IEC (MPEG). Un proyecto reciente del estándar HEVC, denominado "Proyecto de Trabajo HEVC 6" o "WD6", se describe en el documento JCTVC-H1003, Bross y otros, "Proyecto de la especificación del texto de codificación de vídeo de alta eficiencia (HEVC) 6", Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) de UIT-T SG16 WP3 e ISO/IEC JTC 1/SC29/WG11, octava Reunión: San José, California, EE.UU., 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 20 de vídeo y el decodificador 30 de vídeo pueden operar de acuerdo con otros estándares de propiedad o de la industria, tales como el estándar ITU-T H.264, denominado alternativamente como MPEG 4, Parte 10, Codificación Avanzada de Vídeo (AVC), o extensiones de dichos estándares. Las técnicas de esta divulgación, sin embargo, no se limitan a ningún estándar de codificación particular. Otros ejemplos incluyen MPEG-2 y UIT-T H.263.

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

El codificador 20 de vídeo y el decodificador 30 de vídeo, cada uno, puede ser implementado como cualquiera de una variedad de circuitos de codificador adecuados, tal como uno o más microprocesadores, procesadores de señales digitales (DSPs), aplicación de circuitos integrados específicos (ASICs), matrices de puertas programables (FPGAs), lógica discreta, software, hardware, firmware o cualquier fusión de los mismos. Cuando las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio legible por ordenador adecuado, no transitorio, y ejecutar las instrucciones en hardware usando uno o más procesadores para llevar a cabo las técnicas de esta divulgación. Cada uno del codificador 20 de vídeo y el decodificador 30 de vídeo pueden 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 20 de vídeo puede implementar cualquiera o todas las técnicas de esta divulgación para CABAC en un proceso de codificación de vídeo. Del mismo modo, el decodificador 30 de vídeo puede implementar cualquiera o todas estas técnicas para CABAC en un proceso de codificación de vídeo. Un codificador de vídeo, tal como se describe en esta divulgación, puede referirse a un codificador de vídeo o a un decodificador de vídeo. Del mismo modo, una unidad de codificación de vídeo puede referirse a un codificador de vídeo o a un decodificador de vídeo. Del mismo modo, la codificación de vídeo puede referirse a la codificación de vídeo o a la decodificación de vídeo.

En un ejemplo de la divulgación, el codificador 20 de vídeo puede estar configurado para determinar un primer tipo de predicción para un bloque de datos de vídeo en una porción P, que representa el primer tipo de predicción como un elemento de sintaxis de tipo de predicción porción P, determinar un segundo tipo de predicción para un bloque de datos de vídeo en una porción B, representar el segundo tipo de predicción como un elemento de sintaxis de tipo de predicción porción B, determinar una binarización porción P para el elemento de sintaxis de tipo de predicción porción P, determinar una binarización de porción B para el elemento de sintaxis de tipo predicción de porción B, en el que el tipo de predicción del elemento de sintaxis de porción P y el elemento de sintaxis del tipo de predicción de porción B se determinan usando la misma lógica de binarización y codificando los datos de vídeo basados en las binarizaciones del elemento de sintaxis de tipo de predicción de porción P y el elemento de sintaxis de predicción de porción B.

En otro ejemplo de la divulgación, el decodificador 30 de vídeo puede estar configurado para asignar un elemento de sintaxis de tipo de predicción de porción P binarizada a un tipo de predicción usando una asignación de binarización para un bloque de datos de vídeo en una porción P, asignar un elemento de sintaxis de tipo de predicción de porción B binarizada para un tipo de predicción utilizando la misma asignación de binarización para un bloque de datos de vídeo en una porción B, y decodificar los datos de vídeo en base a los tipos de predicción asignados.

En otro ejemplo de la divulgación, el codificador 20 de vídeo puede estar configurado para determinar un tipo de partición para un modo de predicción para un bloque de datos de vídeo, codificar una partición de tipo bin de un elemento de sintaxis de tipo de predicción para el bloque de datos de vídeo usando CABAC con un contexto único, en el que el contexto individual es el mismo para cualquier tipo de partición, y codificar un tamaño de partición bin del elemento de sintaxis de tipo de predicción para el bloque de datos de vídeo utilizando CABAC en el modo de derivación.

En otro ejemplo de la divulgación, el decodificador 30 de vídeo puede estar configurado para recibir un elemento de sintaxis de tipo de predicción para un bloque de datos de vídeo que ha sido codificado usando CABAC, incluyendo el elemento de sintaxis de tipo de predicción una partición de tipo bin que representa un tipo de partición y un tamaño de partición bin que representa un tamaño de partición, decodificar el tipo de partición bin del elemento de sintaxis de tipo de predicción utilizando CABAC con un solo contexto, en el que el contexto individual es el mismo para cualquier tipo de partición, y decodificar el tamaño de partición bin del elemento de sintaxis de tipo de predicción utilizando CABAC en modo de derivación.

En otro ejemplo de la divulgación, el codificador 20 de vídeo y el decodificador 30 de vídeo pueden estar configurados para codificar un indicador de bloque codificado croma Cb de un bloque de datos de vídeo utilizando CABAC, en el que la codificación del indicador de bloque codificado croma Cb comprende el uso de un conjunto de contexto que incluye uno o más contextos como parte de la CABAC, y el código de un indicador de bloque codificado croma Cr utilizando CABAC, en el que la codificación del indicador de bloque codificado croma Cr comprende utilizar el mismo contexto establecido como el indicador de bloque codificado croma Cb como parte de la CABAC.

El JCT-VC está trabajando en la elaboración del estándar HEVC. Los esfuerzos de normalización HEVC se basan en un modelo de evolución de un dispositivo de codificación de vídeo indicado como el Modelo de Prueba HEVC (HM). El HM presume varias capacidades adicionales de dispositivos de codificación de vídeo en relación con los dispositivos existentes de acuerdo con, por ejemplo, UIT-T H.264/AVC. Por ejemplo, aunque H.264 ofrece nueve modos de codificación intra-predicción, el HM puede proporcionar un máximo de treinta y tres modos de codificación intra-predicción. En la siguiente sección se describirán algunos aspectos del HM en más detalle.

En general, el modelo de trabajo del HM describe que un fotograma de vídeo o imagen se puede dividir en una secuencia de bloques de árbol o unidades más grandes de codificación (LCU) que incluyen ambas muestras luminancia y croma. Un bloque de árbol tiene un propósito similar a un macrobloque de la norma H.264. Una porción incluye un número de bloques de árbol consecutivos en orden de codificación. Un fotograma de vídeo o imagen se puede dividir en uno o más sectores. Cada bloque de árbol puede dividirse en unidades de codificación (CUS) de acuerdo con un árbol cuádruple. Por ejemplo, un bloque de árbol, como un nodo raíz del árbol de cuatro ramas, se puede dividir en cuatro nodos hijos, y cada nodo hijo puede a su vez ser un nodo padre y ser dividido en otros cuatro nodos hijos. Un nodo hijo final, de una sola pieza, como un nodo de hoja del árbol de cuatro ramas, comprende un nodo de codificación, es decir, un bloque de vídeo codificado. La sintaxis de los datos asociados con un flujo de bits codificados puede definir un número máximo de veces que un bloque de árbol puede dividirse, y también puede definir un tamaño mínimo de los nodos de codificación.

La CU incluye unas unidades de codificación de nodo y de predicción (PUs) y unidades de transformación (TUs) asociadas con el nodo de codificación. Un tamaño de la CU corresponde generalmente a un tamaño del nodo de codificación y normalmente debe ser de forma cuadrada. El tamaño de la CU puede variar de 8 x 8 píxeles hasta el tamaño del bloque de árbol con un máximo de 64 x 64 píxeles o mayores. Cada CU puede contener uno o más PUs y una o más unidades de formación. Los datos de la sintaxis asociada con una CU pueden describir, por ejemplo, la partición de la CU en una o más PUs. Los modos de particiones pueden ser diferentes entre sí si la CU se omite o se codifica en modo directo, se codifica en modo de intra-predicción, o se codifica en modo de inter-predicción. Las PUs pueden dividirse para no ser de forma cuadrada. Los datos de la sintaxis asociada con una CU también pueden

describir, por ejemplo, la partición de la CU en una o más unidades de formación de acuerdo con un árbol cuádruple. Una TU puede ser cuadrada o ser de forma no cuadrada.

El estándar HEVC emergente permite las transformaciones de acuerdo a TUs, que puede ser diferentes para diferentes CUs. Las TUs están dimensionadas típicamente en función del tamaño de las PUs dentro de una CU determinada definida para un LCU particionada, aunque esto puede no ser siempre el caso. Las TUs suelen ser del mismo tamaño o más pequeño que las PUs. En algunos ejemplos, las muestras residuales correspondientes a una CU se pueden subdividir en unidades más pequeñas utilizando una estructura de árbol cuádruple conocida como "árbol cuádruple residual" (RQT). Los nodos de hoja del RQT pueden ser referidos como unidades de transformación (TUs). Los valores de diferencia de píxeles asociados con las TUs pueden ser transformados para producir coeficientes de transformación, que pueden ser cuantificados.

En general, una PU se refiere a datos relacionados con el proceso de predicción. Por ejemplo, cuando la PU está intra-codificada, la PU puede incluir datos que describen un modo de intra-predicción para la PU. Como otro ejemplo, cuando la PU está inter-codificada, la PU puede incluir 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 procesos de transformación y de cuantificación. Una CU dada que tiene una o más unidades PU también puede incluir una o más unidades de transformación (TUs). Después de la predicción, el codificador 20 de vídeo 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 para hacer referencia a los valores residuales, más que el bloque de vídeo original. Los valores residuales comprenden valores de diferencia de píxeles que pueden ser transformados en coeficientes de transformación cuantificados, y escaneados utilizando las transformadas y otra información de transformación especificada en las TUs para producir coeficientes de transformación serializados para la codificación de entropía. El nodo de codificación puede ser actualizado una vez más para referirse a estos coeficientes de transformación serializados. Esta divulgación suele utilizar el término "bloque de vídeo" para referirse a un nodo de codificación de una CU. En algunos casos específicos, esta divulgación también puede usar el término "bloque de vídeo" para referirse a un bloque de árbol, es decir, LCU, o una CU, que incluye un nodo de codificación y PUs y TUs.

Una secuencia de vídeo incluye típicamente una serie de fotogramas de vídeo o imágenes. Un grupo de imágenes (GOP) comprende generalmente una serie de una o más de las imágenes de vídeo. Un GOP puede incluir datos de sintaxis en una cabecera del GOP, una cabecera de una o más de las imágenes, o en otro lugar, que describe un número de imágenes incluidas en el GOP. Cada porción de una imagen puede incluir datos de sintaxis de porciones que describen un modo de codificación para el respectivo segmento. El codificador 20 de vídeo normalmente funciona en bloques de vídeo dentro de porciones de vídeo individuales para 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 especificado.

Como ejemplo, el HM soporta la predicción en varios tamaños de PU. Suponiendo que el tamaño de una CU particular es $2N \times 2N$, el HM soporta la intra-predicción en tamaños PU de $2N \times 2N$ o $N \times N$, y la inter-predicción en tamaños simétricos de la PU de $2N \times 2N$, $2N \times N$, $N \times 2N$ o $N \times N$. El HM también admite la partición asimétrica para inter-predicción en tamaños 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 está dividida, mientras que la otra dirección se divide en 25% y 75%. La porción de la CU correspondiente a la partición del 25% se indica con una "n", seguido de una indicación de "Arriba", "Abajo", "Izquierda" o "Derecha". Así, por ejemplo, $2N \times nU$ "se refiere a una CU $2N \times 2N$ que está dividida horizontalmente 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 dos tipos de particiones cuadradas y no cuadradas de intra-predicción e inter-predicción. La partición 102 es una partición $2N \times 2N$ y puede ser utilizada para la intra-predicción y la inter-predicción. La partición 104 es una partición de $N \times N$ y puede ser utilizada para la intra-predicción y la inter-predicción. La partición 106 es una partición $2N \times N$ y se utiliza actualmente en HEVC para inter-predicción. La partición 108 es una partición $N \times 2N$ y se utiliza actualmente en HEVC para inter-predicción.

La figura 5 un dibujo conceptual que muestra los tipos de partición asimétricos. La partición 110 es una partición $2N \times nU$ y se utiliza actualmente en HEVC para inter-predicción. La partición 112 es una partición $2N \times nD$ y se utiliza actualmente en HEVC para inter-predicción. La partición 114 es una partición $nL \times 2N$ y se utiliza actualmente en HEVC para inter-predicción. La partición 116 es una partición $nR \times 2N$ y se utiliza actualmente en HEVC para inter-predicción.

En esta divulgación, " $N \times N$ " y " N por N " pueden utilizarse indistintamente para referirse a las dimensiones en 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 16×16 tendrá 16 píxeles en una dirección vertical ($Y = 16$) y 16 píxeles en una

dirección horizontal ($x = 16$). Del mismo modo, un bloque de $N \times N$ generalmente tiene N píxeles en una dirección vertical y N píxeles en dirección horizontal, donde n representa un valor entero no negativo. Los píxeles de un bloque pueden estar dispuestos en filas y columnas. Por otra parte, los bloques no necesitan necesariamente tener el mismo número de píxeles en la dirección horizontal como 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 intra-predictiva o inter-predictiva utilizando las PUs de una CU, el codificador 20 de vídeo puede calcular datos residuales a los que se aplican las transformaciones especificadas por unidades de formación de la CU. Los datos residuales pueden corresponder a diferencias de píxel entre los píxeles de la imagen sin codificar y los valores de predicción correspondientes a la CU. El codificador 20 de vídeo 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 coeficientes de transformación, el codificador 20 de vídeo puede llevar a cabo la cuantificación de los coeficientes de transformación. La cuantificación se refiere generalmente a un proceso en el que se transforman los coeficientes que se cuantifican a la posibilidad de reducir la cantidad de datos utilizados para representar los coeficientes, proporcionando una compresión adicional. El proceso de cuantificación puede reducir la profundidad de bits asociados con algunos o todos de los coeficientes. Por ejemplo, un valor de n bits puede redondearse a la baja a un valor de m bits durante la cuantización, donde n es mayor que m .

En algunos ejemplos, el codificador 20 de vídeo puede utilizar un orden de exploración predefinida para escanear los coeficientes de transformación cuantificados para producir un vector serializado que puede ser entropía codificada. En otros ejemplos, el codificador 20 de vídeo puede realizar una exploración adaptativa. Después de escanear los coeficientes de transformación cuantificados para formar un vector unidimensional, el codificador 20 de vídeo puede codificar la entropía del vector unidimensional, por ejemplo, de acuerdo con la codificación de longitud variable adaptativa según el contexto (CAVLC), el contexto de codificación aritmética binaria adaptativa (CABAC), mediante codificación aritmética binaria adaptativa según el contexto basada en la sintaxis (SBAC), codificación de entropía de partición de intervalos de probabilidades (PIPE) u otra metodología de codificación de entropía. El codificador 20 de vídeo puede también codificar la entropía de elementos de sintaxis asociados con los datos de vídeo codificados para su uso mediante el decodificador 30 de vídeo en la decodificación de los datos de vídeo.

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

Esta divulgación se refiere a técnicas para codificadores de entropía de codificación aritmética binaria adaptativa según el contexto (CABAC) u otros codificadores de entropía, tal como codificación de entropía de partición de intervalos de probabilidad (PIPE) o codificadores relacionados. La codificación aritmética es una forma de codificación de entropía utilizada en muchos algoritmos de compresión que tiene una alta eficiencia de codificación, ya que es capaz de asignar símbolos a palabras de código de longitud no enteros. Un ejemplo de un algoritmo de codificación aritmética es la codificación aritmética binaria adaptativa según el contexto (CABAC) utilizada en H.264/AVC.

En general, la codificación de símbolos de datos usando CABAC implica una o más de las siguientes etapas:

(1) Binarización: Si un símbolo que ha de codificarse no es de valor binario, se asigna a una secuencia de los llamados "bins". Cada bin puede tener un valor de "0" o "1".

(2) Asignación de contexto: Cada bin (en modo regular) se asigna a un contexto. Un modelo de contexto determina cómo se calcula un contexto para un contenedor determinado, basándose en la información disponible para el bin, tal como los valores de los símbolos previamente codificados o número de bin.

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

(4) Actualización de estado: La probabilidad (estado) para un contexto seleccionado se actualiza basándose en el valor real codificado del bin (por ejemplo, si el valor bin era "1", la probabilidad de "1s" se incrementa).

Cabe señalar que la codificación de entropía de partición de intervalos de probabilidad (PIPE) utiliza principios similares a los de la codificación aritmética, y por lo tanto puede también utilizar las técnicas de esta divulgación.

CABAC en H.264/AVC y HEVC utiliza estados, y cada estado está implícitamente relacionado con una probabilidad. Hay variantes de CABAC, en las que una probabilidad de un símbolo ("0" o "1") se utiliza directamente, es decir, la probabilidad (o una versión de número entero de la misma) es el estado. Por ejemplo, tales variantes de CABAC se describen en "Descripción de la propuesta de codificación de tecnología por parte de France Telecom, NTT, NTT DoCoMo, Panasonic y Technicolor" JCTVC-A114, primera Reunión JCT-VC, Dresden, DE, Abril 2010, denominada "JCTVC-A114" en lo sucesivo, y A. Alshin y E. Alshina, "Multi-parameter probability update for CABAC", JCTVC-F254, sexta Reunión JCT-VC, Torino, IT, julio 2011, denominado "JCTVC-F254" en lo sucesivo.

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

En esta divulgación, se propone una reducción en el número de contextos CABAC utilizados para los elementos de sintaxis, *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 insignificantes cambios de eficiencia de codificación. Las reducciones de contexto propuestas para los elementos de sintaxis anteriores se pueden utilizar solos o en cualquier fusión.

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 está codificada en modo inter-predicción. El elemento de sintaxis *pred_mode_flag* igual a 1 especifica que la unidad de codificación actual se codifica en el modo de intra-predicción. 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 candidato de fusión de la lista de candidatos de fusión, donde x0, y0 especifican la ubicación (x0, y0) de la muestra luminancia superior izquierda del bloque de predicción considerado en relación con la muestra luminancia superior izquierda de la imagen. Cuando *merge_idx*[x0] [y0] no está presente, se considera que es igual a 0. Una lista de candidatos de fusión es una lista de las unidades de codificación adyacente a una de las unidades actuales, de la cual se puede copiar la información de movimiento.

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

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

Los elementos de sintaxis *CBF_cb*, *cbf_cr* indican sí o no el croma (Cb y Cr, respectivamente) transforman bloques que 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 la transformada que no son iguales a 0. El índice de matriz x0, y0 especifica la ubicación (x0, y0) de la muestra luminancia superior izquierda del bloque de transformación considerado en relación con la muestra luminancia superior izquierda de la imagen. El índice de matriz *trafoDepth* especifica el nivel de subdivisión actual de una unidad de codificación en bloques para transformar la codificación. El índice de matriz *trafoDepth* es igual a 0 para los bloques que corresponden a las unidades de codificación. Cuando *cbf_cb*[x0] [y0] [trafoDepth] no está presente y el modo de predicción no es intra-predicción, el valor de *CBF_cb*[x0] [y0] [trafoDepth] se considera que 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 la transformada que no son iguales a 0. El índice de matriz x0, y0 especifica la ubicación (x0, y0) de la muestra luminancia superior izquierda del bloque de transformación considerado en relación con la muestra luminancia 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 fin de transformar la codificación. El índice de matriz *trafoDepth* es igual a 0 para los bloques que corresponden a las unidades de codificación. Cuando *cbf_cr* [x0] [y0] [trafoDepth] no está presente y el modo de predicción no es intra-predicción, el valor de *cbf_cr* [x0] [y0] [trafoDepth] se considera que es igual a 0.

El elemento de sintaxis *Coef_abs_nivel_greater1_flag* especifica la posición de exploración n si hay que transformar los niveles de coeficiente superior a 1. Cuando *coeff_abs_level_greater1_flag*[n] no está presente, se considera que es igual a 0.

El elemento de sintaxis *coeff_abs_level_greater2_flag* especifica para la posición de exploración n si hay transformar niveles de coeficiente mayor que 2. Cuando *coeff_abs_level_greater2_flag*[n] no está presente, se considera que es

igual a 0.

5 En una propuesta de HEVC, diferentes binarizaciones en el elemento de sintaxis *pred_type* se utilizan en porciones P y B como se muestra en la Tabla 1. Esta divulgación propone el uso de las mismas binarizaciones para porciones P y B. Los ejemplos se muestran en las Tablas 2-4. La Tabla 5 muestra el impacto en el rendimiento de codificación de la porción P en condiciones de ensayo comunes (por ejemplo, véase F. Bossen, "Common test conditions and software reference configurations" JCTVC-F900).

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

Tipo de Porción	Valor de <i>pred_type</i>	PredMode	PartMode	Cadena de Bin	
				cLog2CUSize > Log2MinCUSize	cLog2CUSize = = Log2MinCUSize
I	0	MODO_INTRA	PARTE_2Nx2N	-	1
	1	MODO_INTRA	PARTE_NxN	-	0
	0	MODO_INTER	PARTE_2Nx2N	0 1	0 1
	1	MODO_INTER	PARTE_2NxN	0 0 1 1	0 0 1
	2	MODO_INTER	PARTE_Nx2N	0 0 0 1 1	0 0 0 1
	4	MODO_INTER	PARTE_2NxN _U	0 0 1 0 0	-
	5	MODO_INTER	PARTE_2NxN _D	0 0 1 0 1	-
P	6	MODO_INTER	PARTE_nLx2N	0 0 0 1 0 0	-
	7	MODO_INTER	PARTE_nRx2N	0 0 0 1 0 1	-
	3	MODO_INTER	PARTE_NxN	-	0 0 0 0
	4	MODO_INTRA	PARTE_2Nx2N	1	1 1
	5	MODO_INTRA	PARTE_NxN	-	1 0
	0	MODO_INTER	PARTE_2Nx2N	1	1
	1	MODO_INTER	PARTE_2NxN	0 1 1	0 1
	2	MODO_INTER	PARTE_Nx2N	0 0 1 1	0 0 1
	4	MODO_INTER	PARTE_2NxN _U	0 1 0 0	-
	5	MODO_INTER	PARTE_2NxN _D	0 1 0 1	-
	6	MODO_INTER	PARTE_nLx2N	0 0 1 0 0	-
B	7	MODO_INTER	PARTE_nRx2N	0 0 1 0 1	-
	3	MODO_INTER	PARTE_NxN	-	0 0 0 1
	4	MODO_INTRA	PARTE_2Nx2N	0 0 0	0 0 0 0
	5	MODO_INTRA	PARTE_NxN	-	0 0 0 1
	0	MODO_INTER	PARTE_2Nx2N	1	1
	1	MODO_INTER	PARTE_2NxN	0 1 1	0 1
	2	MODO_INTER	PARTE_Nx2N	0 0 1 1	0 0 1
	4	MODO_INTER	PARTE_2NxN _U	0 1 0 0	-
	5	MODO_INTER	PARTE_2NxN _D	0 1 0 1	-
	6	MODO_INTER	PARTE_nLx2N	0 0 1 0 0	-
	7	MODO_INTER	PARTE_nRx2N	0 0 1 0 1	-
3	MODO_INTER	PARTE_NxN	-	0 0 0 1	
4	MODO_INTRA	PARTE_2Nx2N	0 0 0	0 0 0 0	
5	MODO_INTRA	PARTE_NxN	-	0 0 0 1	

- 5 Como puede verse en la Tabla 1, LAS porciones I (por ejemplo, porciones que sólo incluyen bloques intra-predichos), incluyen dos tipos de predicción diferentes (*pred_type*). Una cadena de bin (binarización) se utiliza para un bloque intra-predicho con un tipo de partición $2N \times 2N$, y otra cadena bin se utiliza para un bloque intra-predicho con un tipo de partición $N \times N$. Como se muestra en la Tabla 1, la cadena bin utilizada para las porciones I no depende del tamaño de la CU.
- Para las porciones P y B, en la Tabla 1, diferentes cadenas bin se utilizan para cada valor de *pred_type*. Una vez más, el valor de *pred_type* depende del modo de predicción (inter-predicción o intra-predicción) y del tipo de partición utilizada. Para porciones P y B, la cadena bin real utilizada depende también del tamaño de la CU que está codificada y si está habilitada o no la inter-predicción para un tamaño de bloque 4×4 .
- 10 La primera columna bajo la cadena bin se aplica para la situación en la que la función logarítmica de tamaño de CU de la CU que se está codificando es mayor que la función logarítmica del tamaño mínimo permisible de la CU. Según un ejemplo en HEVC, la primera columna de cadenas bin se utiliza si $cLog2CUSize > Log2MinCUsize$. La función logarítmica se utiliza para crear un número más pequeño, de manera que puede ser utilizado un índice consecutivo más pequeño.
- 15 Si la función logarítmica del tamaño de CU de la CU se está codificando es equivalente a la función logarítmica del tamaño mínimo CU permisible (es decir, $cLog2CUSize = Log2MinCUsize$), entonces una de las columnas 2 y 3 en virtud de la cadena bin en la Tabla 1 se utiliza para seleccionar la binarización. La columna 2 se utiliza cuando la función logarítmica del de tamaño CU de la CU que se está codificando es equivalente a 3 y la inter-predicción para una CU 4×4 no está habilitada (es decir, $cLog2CUSize = 3 \ \&\&!inter_4x4_enabled_flag$). La columna 3 se utiliza
- 20 cuando la función logarítmica del tamaño de CU de la CU se está codificando es mayor que 3 o cuando la inter-predicción para una CU 4×4 está activada (es decir, $cLog2CUSize > 3 \ || \ inter_4x4_enabled_flag$).
- La tabla 2 a continuación muestra un ejemplo de binarizaciones donde las porciones P y B utilizan las mismas cadenas bin, de acuerdo con uno o más ejemplos descritos en esta divulgación. Como se muestra en la Tabla 2, las porciones P utilizan las mismas binarizaciones utilizadas para los segmentos B en la Tabla 1. De esta manera, no es necesario almacenar y utilizar un conjunto separado de contextos para ambas porciones P y B. Como tal, el número
- 25 total de contextos necesario para codificar el *pred_type* se reduce al elemento de sintaxis. Por otra parte, sólo una asignación (en lugar de dos) entre la lógica de cadena bin (que se muestra en las columnas (1) - (3)) y la cadena bin actual necesita almacenarse.

Tabla 2 Binarización para pred_type en un ejemplo de la divulgación

Tipo de Porción	Valor de pred_type	PredMode	PartMode	Cadena bin	
				cLog2CUSize> Log2MinCUSize	cLog2CUSize = = Log2MinCUSize
I	0	MODO_INTRA	PARTE_2Nx2N	-	1
	1	MODO_INTRA	PARTE_NxN	-	0
P o B	0	MODO_INTER	PARTE_2Nx2N	1	1
	1	MODO_INTER	PARTE_2NxN	011	01
	2	MODO_INTER	PARTE_Nx2N	0011	001
	4	MODO_INTER	PARTE_2NxN _U	0100	-
	5	MODO_INTER	PARTE_2NxN _D	0101	-
	6	MODO_INTER	PARTE_nLx2N	00100	-
	7	MODO_INTER	PARTE_nRx2N	00101	-
	3	MODO_INTER	PARTE_NxN	-	0001
	4	MODO_INTRA	PARTE_2Nx2N	000	000 0
	5	MODO_INTRA	PARTE_NxN	-	000 1
					cLog2CUSize> 3 inter_4x4_enabled_flag
					1
					0
					1
					01
					001
					-
					-
					-
					0001
					0000 0
					0000 1

La Tabla 3 a continuación muestra otro ejemplo de una binarización para *pred_type*. En este ejemplo, las porciones B utilizan las mismas binarizaciones que las porciones P en la Tabla 1. La Tabla 4 a continuación muestra un ejemplo adicional de porciones P y porciones B utilizando las mismas binarizaciones. Las Tablas 2-4 sólo están hechas para mostrar ejemplos de binarizaciones compartidas entre porciones P y B. Cualquier binarización o regla de binarización puede utilizarse, de manera que los elementos de sintaxis *pred_type* para ambas porciones P y B comparten las mismas binarizaciones.

El codificador 20 de vídeo y el decodificador 30 de vídeo pueden almacenar las mismas reglas de asignación y tablas de asignación (por ejemplo, como se muestra en las Tablas 2-4) para su uso con porciones P y B. La codificación y decodificación CABAC puede aplicarse al elemento de sintaxis *pred_type* utilizando estas asignaciones.

De esta manera, el codificador de vídeo 20 puede estar configurado para determinar un primer tipo de predicción para un bloque de datos de vídeo en una porción P, que representa el primer tipo de predicción como un elemento de sintaxis de tipo de predicción de porción P, determinar un segundo tipo de predicción para un bloque de los datos de vídeo en una porción B, representar el segundo tipo de predicción como un elemento de sintaxis de tipo de predicción de porción B, determinar una binarización de porción P para el elemento de sintaxis de tipo de predicción de porción P, determinar una binarización de porción B para el elemento de sintaxis de tipo de predicción de porción B, en el que el elemento de sintaxis de tipo de predicción de porción P y el elemento de sintaxis de tipo de predicción de porción B se determinan usando la misma lógica de binarización y codificar los datos de vídeo basado en las binarizaciones del elemento de sintaxis de tipo de predicción de porción P y el elemento de sintaxis de predicción de porción B.

El codificador 20 de vídeo puede estar configurado además para binarizar el elemento de sintaxis de tipo de predicción de porción P con la binarización de porción P decidida, binarizar el elemento de sintaxis de tipo de predicción de porción B con la binarización de porción B decidida, aplicar una codificación aritmética binaria adaptativa según el contexto (CABAC) al elemento de sintaxis tipo de predicción binarizada de porción P, y aplicar codificación aritmética binaria adaptativa según el contexto (CABAC) al elemento de sintaxis de tipo de predicción binarizada de porción B.

Del mismo modo, el decodificador 30 de vídeo puede estar configurado para asignar un elemento de sintaxis de tipo de predicción binarizada de porción P a un tipo de predicción usando una asignación de binarización para un bloque de datos de vídeo en una porción P, asignar un elemento de sintaxis de tipo de predicción binarizada de porción B a un tipo de predicción utilizando la misma asignación de binarización para un bloque de datos de vídeo en una porción B, y decodificar los datos de vídeo en base a los tipos de predicción asignados.

El decodificador de vídeo 30 puede estar configurado además para recibir un elemento de sintaxis de tipo de predicción de porción P codificado aritmético binario adaptativo según el contexto que indica el tipo de predicción para el bloque de datos de vídeo en una porción P, recibir un elemento de sintaxis de tipo de predicción de porción B codificado aritmético binario adaptativo según el contexto que indica el tipo de predicción para el bloque de datos de vídeo en una porción B, decodificar el elemento de sintaxis de tipo de predicción de porción P para producir el elemento de sintaxis de tipo de predicción porción P binarizado, y decodificar el elemento de sintaxis tipo de predicción de porción B para producir el elemento de sintaxis de tipo de predicción de porción B binarizado.

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

Tipo de Porción	Valor de <i>pred_type</i>	PredMode	PartMode	Cadena bin	
				cLog2CUSize>Log2MinCUSize	cLog2CUSize = Log2MinCUSize
1	0	MODO_INTRA	PARTE_2Nx2N	-	cLog2CUSize = 3 &&inter_4x4_enabled_flag
	1	MODO_INTRA	PARTE_NxN	-	inter_4x4_enabled_flag
P o B	0	MODO_INTER	PARTE_2Nx2N	0 1	cLog2CUSize> 3
	1	MODO_INTER	PARTE_2NxN	0011	inter_4x4_enabled_flag
	2	MODO_INTER	PARTE_Nx2N	00011	0 001
	4	MODO_INTER	PARTE_2NxN	0 0100	-
	5	MODO_INTER	PARTE_2NxN	0 0101	-
	6	MODO_INTER	PARTE_nLx2N	0 00100	-
	7	MODO_INTER	PARTE_nRx2N	0 00101	-
	3	MODO_INTER	PARTE_NxN	-	0000
	4	MODO_INTRA	PARTE_2Nx2N	11	11
	5	MODO_INTRA	PARTE_NxN	10	10

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

Tipo de Porción	Valor de <i>pred_type</i>	PredMode	PartMode	Cadena bin	
				cLog2CUSize>Log2MinCUSize	cLog2CUSize = = Log2MinCUSize
I	0	MODO_INTRA	PARTE_2Nx2N	-	1
	1	MODO_INTRA	PARTE_NxN	-	0
P o B	0	MODO_INTER	PARTE_2Nx2N	1	1
	1	MODO_INTER	PARTE_2NxN	01	01
	2	MODO_INTER	PARTE_Nx2N	001	001
	4	MODO_INTER	PARTE_2NxN	0100	-
	5	MODO_INTER	PARTE_2NxN	0101	-
	6	MODO_INTER	PARTE_nLx2N	0000	-
	7	MODO_INTER	PARTE_nRx2N	0001	-
	3	MODO_INTER	PARTE_NxN	-	000
	4	MODO_INTRA	PARTE_2Nx2N	000	000 0
	5	MODO_INTRA	PARTE_NxN	-	000 1
					cLog2CUSize = = 3 &&inter_4x4_enabled_flag

La Tabla 5 a continuación muestra el rendimiento de codificación utilizando la binarización compartida para las porciones P y B que se muestran en la Tabla 2. Como puede verse en la Tabla 5, poca o ninguna eficacia de la codificación se pierde usando las binarizaciones compartidos. P HE (alta eficiencia) de bajo retardo es una condición de prueba común para binarizaciones de porciones predichas (P) unidireccionalmente. Las clases A-E representan diferentes resoluciones de marco. 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 720p. Un cambio de porcentaje de 0,1 a 0.2 en la condición de prueba P HE de bajo retardo se considera generalmente que es insignificante.

Índice BD	P HE de bajo retardo		
	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%
Todas	0,01%	0,04%	0,03%
Enc T[%]			
Dec T[%]			

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

Opcionalmente, las mismas binarizaciones (no limitadas a las Tablas 2-4) para el tipo de predicción (incluye el tamaño de predicción y/o el modo de predicción) se pueden compartir en dos y más tipos diferentes de porciones de inter-predicción. Las porciones de inter-predicción pueden incluir, pero no están limitadas a:

- a. Porción P: porción que sólo soporta predicción de movimiento unidireccional
- b. Porción B: porción que soporta predicción de movimiento unidireccional y bidireccional
- c. En codificación de vídeo escalable: la capa de mejora puede compartir las mismas binarizaciones con la capa base.
- d. En la codificación de múltiples vistas: vistas diferentes pueden compartir las mismas binarizaciones.

Cuando la partición asimétrica está habilitada, cuatro contextos, igualmente divididos en dos conjuntos de contexto, se utilizan para CABAC en los dos últimos bins para la señalización del elemento de sintaxis *pred_type* para particiones asimétricas (es decir, PART_2NxnU, PART_2NxnD, PART_nLx2N, PART_nRx2N). Dependiendo de si la partición se divide a lo largo de la dirección horizontal o de la dirección vertical, se aplica un conjunto de contexto. El segundo al último bin (es decir, el bin de tipo de partición; *part_mode*) especifica si la CU actual tiene particiones simétricas o particiones asimétricas. El último bin (es decir, el de tamaño de la partición bin; *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 segundo al último (tipo de partición) y el último (tamaño de partición) para el elemento de sintaxis *pred_type*.

Tabla 6.Contextos de los Últimos Dos compartimientos del *Pred_Type*Sintaxis Elemento

Bin	Contexto
Tipo de partición (simétrica o asimétrica)	Conjunto de contexto 1 (2 Contextos, uno para la partición vertical, 1 para la partición horizontal)
Tamaño de la partición (Primera partición es ¼ CU o ¾ CU)	Conjunto de contexto 2 (2 Contextos, uno para ¼ CU y otro para ¾ CU)

Esta divulgación se propone utilizar un contexto para el segundo bin al último (es decir, el tipo de partición bin) y utilizar el modo de derivación en el último bin (es decir, el tamaño de partición bin). Como resultado, el número de contextos se redujo de 4 a 1. La Tabla 7 muestra un ejemplo de contexto utilizado 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 los marcos de acceso aleatorio. B HE de bajo retardo es una condición de prueba que permite la predicción bidireccional.

Tabla 7.Contextos de los últimos dos bins del elemento de sintaxis *Pred_Type* de acuerdo con un ejemplo de esta divulgación.

Bin	Contexto
Tipo de partición (simétrica o asimétrica)	Conjunto de contexto 1 (1 Contexto)
Tamaño de la partición (Primera partición es ¼ CU o ¾ CU)	Modo de derivación (sin contextos)

Índice BD	Todo HE Intra	Acceso aleatorio HE	B HE bajo retardo
-----------	---------------	---------------------	-------------------

	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%
Case E							0,00%	0,30%	0,39%
Todos				0,01%	-0,04%	-0,07%	0,00%	0,06%	0,01%
Enc T[%]									
Dec T[%]									

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

De esta manera, de acuerdo con este ejemplo, el codificador 20 de vídeo puede estar configurado para determinar un tipo de partición para un modo de predicción para un bloque de datos de vídeo, codificar un tipo de partición bin de un elemento de sintaxis de tipo de predicción para el bloque de datos de vídeo usando la codificación aritmética binaria adaptativa según el contexto con un solo contexto, en el que el contexto individual es el mismo para cualquier tipo de partición, y codificar un tamaño de partición bin de una sintaxis de tipo de predicción para el bloque de datos de vídeo usando la codificación aritmética binaria adaptativa según el contexto en el modo de derivación.

Del mismo modo, según este ejemplo, el decodificador 30 de vídeo puede estar configurado para recibir un elemento de sintaxis de tipo de predicción para un bloque de datos de vídeo que ha sido codificado usando la codificación aritmética binaria adaptativa según el contexto (CABAC), incluyendo el elemento de sintaxis de tipo de predicción un tipo de partición bin que representa un tipo de partición y un tamaño de partición bin que representa un tamaño de partición, decodificar el tipo de partición bin del elemento de sintaxis de tipo de predicción usando la codificación aritmética binaria adaptativa según el contexto con un solo contexto, en el que el contexto individual es el mismo para cualquier tipo de partición, y la decodificación del tamaño de la partición bin de la sintaxis de tipo de predicción para un bloque de datos de vídeo utilizando la codificación aritmética binaria adaptativa según el contexto en modo de derivación.

En otro ejemplo, cuando la codificación de un tipo de partición de rectángulo, el modo de derivación o un único contexto se puede utilizar para el bin que indica si el modo de partición es PART_nLx2N o PART_nRx2N, o si el modo es PART_2NxnU, PART_2NxnD. El uso del modo de derivación o un solo contexto es aplicable debido a que la probabilidad de cualquiera de los modos de partición que se utiliza es cercana al 50%. También opcionalmente, el modo de derivación o un único contexto se pueden utilizar para el bin que indica si el modo es una partición simétrica o una partición asimétrica.

El siguiente ejemplo de la divulgación se refiere a la señalización en un modo de "fusión" de inter-predicción. En el modo de fusión, el codificador indica a un decodificador, a través de flujo de bits de señalización de la sintaxis de predicción, copiar un vector de movimiento, el índice de referencia (la identificación de una imagen de referencia, en una lista de imágenes de referencia dado, al 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 el marco de referencia temporal precede o sigue al marco actualmente) a partir de un vector de movimiento candidato seleccionado para una porción circular de la imagen que se va a codificar. Esto se logra mediante la señalización en la corriente de bits de un índice en una lista de vectores de movimiento candidato que identifica el vector de movimiento candidato seleccionado (es decir, el predictor del vector de movimiento espacial en particular (MVP) candidato o candidato MVP temporal).

Por lo tanto, para el modo de fusión, la sintaxis de predicción puede incluir una señal que identifica el modo (en este caso modo "fusión") y un índice (*merge_idx*) que identifica el vector de movimiento candidato seleccionado. En algunos casos, el vector de movimiento candidato estará en una parte causal en referencia a la parte actual. Es decir, el vector de movimiento candidato ya habrá sido decodificado por el decodificador. Como tal, el decodificador ya ha recibido y/o determinado el vector de movimiento, el índice de referencia, y la dirección de predicción de movimiento para la parte causal. Como tal, el decodificador puede simplemente recuperar el vector de movimiento, el índice de referencia, y la dirección de predicción de movimiento asociada con la parte causal de la memoria y copiar estos valores como la información de movimiento para la porción actual. Para reconstruir un bloque en el modo de fusión, el decodificador obtiene el bloque predictivo utilizando la información de movimiento derivada de la porción actual, y añade los datos residuales al bloque predictivo para reconstruir el bloque codificado.

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

Tabla 9. Contextos para los últimos dos bins del elemento de sintaxis *Pred_Type*

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

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

Índice BD	Todo HE Intra			Acceso aleatorio HE			B HE bajo retardo		
	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%
Case E							-0,14%	0,08%	0,78%
Todos				0,01%	-0,05%	-0,02%	-0,01%	-0,09%	0,17%
Enc T[%]									
Dec T[%]									

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

- 5
- Opcionalmente, más de un contexto puede ser utilizado en la codificación de índice de fusión, con algunos bins que comparten el mismo contexto y algunos bins utilizando otros contextos. Como un ejemplo, sólo los bins consecutivos comparten el mismo contexto. Por ejemplo, bin2 y bin3 pueden compartir un contexto; bin2 y bin4 no pueden compartir el mismo contexto, a menos que bin3 también esté compartiendo el contexto.
- 10 Como otro ejemplo, supongamos que el número total de bins del índice de fusión es N (el primer bin es bin0, el último es bin N-1). Y umbrales $thres_i$, $u = 1, \dots, y$, se utilizan para determinar el contexto de compartición en la codificación del índice de fusión. En este ejemplo, las siguientes reglas indican cómo se comparten los contextos entre bins:
1. $0 < Y < N$ (hay menos umbrales que bins)
 - 15 2. $thres_i < thres_{i+1}$
 3. $0 < thres_1$
 4. $thres_y = N$
 5. bin_i compartirá un contexto donde $i = \{thres_y, \dots, thres_{i+1}-1\}$
- Sobre la base de estas reglas, el procedimiento anterior en el que un contexto se utiliza repetidamente en los cuatro bins, se puede ver como un caso en que $N = 4$, $Y = 1$, $thres_1 = 4$. Por lo tanto, bin 0 a bin 3 están compartiendo el mismo contexto.
- Otro ejemplo incluye el establecimiento de $N = 4$, $Y = 2$, $thres_1 = 2$, $thres_2 = 4$. En este ejemplo, bin0 y bin1 comparten los mismos contextos y bin2 y bin3 comparten los mismos contextos.
- 25 La señal inter-predicción (*inter_pred_flag*) especifica si uni-predicción, o bi-predicción se utiliza para la PU actual. En algunos ejemplos, el índice de contexto para la señal inter-predicción es igual a la profundidad de CU actual. Como hay cuatro posibles profundidades de CU (0 a 3), hay cuatro contextos posibles para la codificación de *inter_pred_flag*.
- 30 Esta divulgación propone que el índice de contexto utilizado para seleccionar un contexto para la codificación de la *inter_pred_flag* sea igual a la profundidad de la CU actual (por ejemplo, la descomposición del nivel del árbol cuádruple para CUs), pero tiene un tope de un umbral elegido (es decir, es el menor de la profundidad de la CU actual o un umbral). El umbral puede elegirse para que sea 2 en un ejemplo. Alternativamente, el índice de contexto puede ser igual a la profundidad máxima de la CU menos la profundidad de la CU actual y un tope de un umbral elegido. Alternativamente, una tabla de asignación predefinida se puede diseñar para seleccionar el índice de contexto por una profundidad de CU dada. La tabla de asignación se puede implementar como un conjunto de
- 35 lógica. Como resultado, 3 contextos se usan para codificar el elemento de sintaxis *inter_pred_flag*.
- La Tabla 11 muestra el rendimiento de codificación cuando se cambia la tabla de inicialización, pero el número de contextos no se cambia. La Tabla 12 muestra el rendimiento de codificación de la técnica propuesta que reduce el número de contextos de 4 a 3.

Índice BD	Todo HE Intra			Acceso aleatorio HE			B HE bajo retardo		
	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%
Case E							0,10%	-0,03%	0,65%
Todos				-0,01%	-0,07%	-0,02%	0,00%	0,07%	0,14%
Enc T[%]									
Dec T[%]									

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

Índice BD	Todo HE Intra			Acceso aleatorio HE			B HE bajo retardo		
	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%
Case E							-0,12%	0,08%	0,45%
Todos				0,01%	-0,07%	-0,03%	-0,04%	-0,01%	0,03%
Enc T[%]									
Dec T[%]									

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

- 5 El índice del marco de referencia (*ref_idx_lx*) se señala mediante el uso de un código unario truncado con respecto al marco de referencia activo en la lista asociada (por ejemplo, Lista 0 o Lista 1). Tres contextos se utilizan para codificar el índice de marco de referencia. Un contexto para bin 0, un contexto para bin 1, y un contexto se utiliza en el resto de los bins. La Tabla 13 muestra un ejemplo de las asignaciones de contexto para los bins del código unario para *ref_idx_lx*.

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

Bins de código unario <i>ref_idx_lx</i>	Contexto
Bin 0	Contexto 1
Bin 1	Contexto 2
Bins 2-N (N es el número total de bins)	Contexto 3

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

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

Bins de código unario <i>ref_idx_lx</i>	Contexto
Bin 0	Contexto 1
Bins 1-N (N es el número total de bins)	Contexto 2

Índice BD	Todo HE Intra			Acceso aleatorio HE			B HE bajo retardo		
	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%	0,03%	-0,04%	-0,14%	-0,07%
Clase D				0,03%	0,06%	0,11%	-0,06%	0,19%	-0,09%
Case E							-0,06%	-0,34%	0,48%
Todos				0,00%	-0,01%	-0,01%	-0,03%	-0,09%	0,05%
Enc T[%]									
Dec T[%]									

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

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

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

Indicador de bloque codificado 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 utilizado en cada conjunto todavía puede ser igual a la profundidad de transformación actual asociada con el indicador de bloque croma codificado que está siendo codificada. La Tabla 17 muestra los conjuntos de contexto para los indicadores de bloques codificados croma *cbf_cb* y *cbf_cr* de acuerdo con los ejemplos de esta divulgación. La Tabla 18 muestra el rendimiento de codificación asociado de las modificaciones propuestas.

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

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

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

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

De esta manera, de acuerdo con este ejemplo, el codificador 20 de vídeo y el decodificador 30 de vídeo pueden configurarse para codificar un indicador de bloque croma codificado de Cb para un bloque de datos de vídeo usando la codificación aritmética binaria adaptativa según el contexto (CABAC), en el que CABAC utiliza un conjunto de contexto que incluye uno o más contextos, y el código del indicador de bloque croma codificado Cr utiliza CABAC, en el que CABAC utiliza el mismo conjunto de contexto que el indicador de bloque Cb croma codificado. El codificador 20 de vídeo y el decodificador 30 de vídeo pueden estar configurados además para seleccionar un contexto a partir de los uno o más contextos basados en una profundidad de transformación de una unidad de transformación asociada con el bloque de datos de vídeo.

En una propuesta de HEVC, hay doce conjuntos de contexto, para *coeff_abs_level_greater1_flag* y *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 son igualmente asignados para la luminancia y el componente croma, es decir, 6 conjuntos de contexto para luminancia y 6 contextos de croma. Cada conjunto de contexto consta de 5 contextos. El índice del conjunto de contexto, *ctxSet*, se selecciona basándose en el *coeff_abs_level_greater1_flag* anterior. Para *coeff_abs_level_greater1_flag*, el índice del contexto dentro de un contexto establecido, *greater1Ctx*, se determina con base a los que se arrastran en un máximo de 4. El índice de contexto se puede representarse como:

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

Para *coeff_abs_level_greater2_flag*, el índice del contexto dentro de un contexto establecido, *greater2Ctx*, se basa en el número de *coeff_abs_level_greater1_flag* sea de 1 a un máximo de 4. El índice de contexto se puede

representar como:

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

greater1Ctx basado en el número de los coeficientes significativos y el número de los coeficientes que son mayores que 1. Por otro lado, greater2Ctx basado en el número de los coeficientes que son mayores que 1.

5 En algunos ejemplos, un número diferente de contextos puede ser utilizado en diferentes conjuntos de contexto, incluyendo, por ejemplo:

1. Los conjuntos de contexto para el nivel superior a 1 o para el nivel superior a 2 podrían tener diferente número 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.

10 2. Los conjuntos de contexto para el coeficiente de luminancia pueden tener diferente número de contextos en comparación con el conjunto de contexto para el componente croma. Por ejemplo, el conjunto de contexto 0 para luminancia puede tener 5 contextos y el conjunto de contexto 0 para croma podría tener 4 contextos.

15 3. El conjunto de contexto para el nivel superior a 1 puede tener diferente número de contextos que el conjunto de contexto para el nivel superior a 2. Por ejemplo, el conjunto de contexto 0 para el nivel mayor que 1 podría tener 5 contextos y el conjunto de contexto 0 para el nivel superior a 2 sólo puede tener 3 contextos.

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

1. Los conjuntos de contexto para el coeficiente de luminancia pueden tener diferente número de conjuntos de contexto que los conjuntos de contexto utilizados para el componente croma. Por ejemplo, la luminancia podría utilizar 6 contextos y el croma podría utilizar 4 contextos.

20 2. Los conjuntos de contexto mayores que 1 pueden tener diferente número de conjuntos de contexto que los conjuntos de contexto utilizados superiores a 2. Por ejemplo, el mayor que 1 podría utilizar 6 contextos y el mayor que 2 podría utilizar 4 contextos.

25 Opcionalmente, una métrica se utiliza para determinar qué contexto se utiliza en un conjunto de contexto y el rango de valor de la métrica es mayor que el número de los contextos en el conjunto de contexto. En uno de tales aspectos, un contexto podría estar asociado a uno o más valores de la métrica. El intercambio de contextos se limita preferiblemente a valores continuos. Por ejemplo, si el valor de la métrica es $y = 2$ se asocia al contexto 3, e $y = 1$ e $y = 4$ también pueden asociarse con el contexto 3. Sin embargo, si $y = 3$ está asociada con el contexto 4, $y = 4$ no se puede asociar al contexto 3.

30 Por ejemplo, para `coeff_abs_level_greater1_flag`, los conjuntos de contexto 0 y 3 tienen 5 contextos y los conjuntos de contexto 1, 2, 4 y 5 tienen 2 contextos. Para `coeff_abs_level_greater2_flag`, los conjuntos de contexto 0, 1 y 2 tienen 5 contextos y los conjuntos de contexto 3, 4 y 5 con 2 contextos. Se puede representar como:

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

35 Si `ctxSet = 0` o `ctxSet = 3`, `Thres_greater1 = 4`;
de lo contrario, `Thres_greater1 = 1`.

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

si `ctxSet < 3`, `Thres_greater2 = 4`;
de lo contrario, `Thres_greater2 = 1`

`Thres_greater1` y `Thres_greater2` pueden elegirse de manera diferente en función de las siguientes situaciones:

- 40
1. Componente de luminancia o croma
 2. Los conjuntos de contexto

Como otro ejemplo, para `coeff_abs_level_greater1_flag`, los conjuntos de contexto 0 y 3 tienen 5 contextos y los conjuntos de contexto 1, 2, 4 y 5 tienen 3 contextos. Para `coeff_abs_level_greater2_flag`, los conjuntos de contexto 0, 1 y 2 tienen 5 contextos y los conjuntos de contexto 3, 4 y 5 tienen 2 contextos. Se puede representar como:

45

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

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

En estos ejemplos, el mapa 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 CABAC de `coeff_abs_level_greater_flag` y `coeff_abs_level_greater2_flag` también se modifican para los conjuntos de contexto para `Thres_greater1` o `Thres_greater2` igual a 1. Las modificaciones mueven la inicialización del quinto contexto con antelación para la inicialización del segundo contexto. Este procedimiento propuesto reduce el número de los contextos de 120 a 78.

Indice BD	Todo HE Intra			Acceso aleatorio HE			B HE bajo retardo		
	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%
Case E	0,00%	0,02%	0,03%				0,09%	-0,52%	0,16%
Todos	0,00%	0,01%	0,01%	0,02%	0,15%	-0,09%	0,01%	-0,02%	0,04%
Enc T[%]									
Dec T[%]									

Tabla 21. Rendimiento de codificación del método propuesto en `coeff_abs_level_greater1_flag` y `coeff_abs_level_greater2_flag`.

La Tabla 21 muestra el número de los contextos para todos los elementos de sintaxis mencionados en los apartados anteriores. La reducción total es de 56 contextos.

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

# 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
Cantidad	151	95

La figura 2 es un diagrama de bloques que ilustra un codificador 20 de vídeo de ejemplo que puede aplicar las técnicas descritas en esta divulgación. El codificador 20 de vídeo puede realizar intra e inter-codificación de bloques de vídeo dentro de porciones de vídeo. La intra-codificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial en el vídeo dentro de un marco de vídeo o imagen dado. La inter-codificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal en vídeo dentro de los marcos adyacentes o imágenes de una secuencia de vídeo. El intra-modo (modo I) puede referirse a cualquiera de los diversos modos de compresión basados espaciales. Los inter-modos, tales como la predicción unidireccional (modo P) o bi-predicción (modo B), pueden referirse a cualquiera de los diversos modos de compresión basados temporales.

En el ejemplo de la figura. 2, el codificador 20 de vídeo 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 de entropía 56. La unidad de predicción 41 incluye una unidad de estimación de movimiento 42, una unidad de compensación de movimiento 44, y la unidad de predicción intra 46. Para bloquear la reconstrucción de vídeo, el codificador 20 de vídeo también incluye una unidad de cuantificación inversa 58, una unidad transformada inversa 60, y un sumador 62. Un filtro de desbloqueo (que no se muestra en la figura 2) puede incluirse también para filtrar límites de bloque para eliminar artefactos de formación de bloques de vídeo reconstruido. Si se desea, el filtro de desbloqueo sería típicamente un filtrado del salida del sumador 62. Filtros de bucle adicionales (en lazo o bucle de mensaje) también se pueden usar además del filtro de desbloqueo.
- Como se muestra en la figura 2, el codificador 20 de vídeo recibe datos de vídeo, y la unidad de partición 35 parte los datos en bloques de vídeo. Esta partición puede incluir partición en porciones, segmentos, u otras unidades más grandes, como también particiones de bloques de vídeo, por ejemplo, de acuerdo con una estructura de árbol cuádruple de LCUs y CUs. El codificador 20 de vídeo ilustra en general los componentes que codifican los bloques de vídeo dentro de una porción vídeo a codificar. La porción puede dividirse en múltiples bloques de vídeo (y posiblemente en conjuntos de bloques de vídeo que se hace referencia como segmentos). La unidad de predicción 41 puede seleccionar uno de una pluralidad de posibles modos de codificación, tal como uno de una pluralidad de modos de codificación intra o una de una pluralidad de modos de codificación inter, para el bloque de vídeo actual basándose en los resultados de errores (por ejemplo, velocidad de codificación y nivel de distorsión). La unidad de predicción 41 puede proporcionar el bloque intra o inter-codificado resultante al sumador 50 para generar datos de bloque residuales y al sumador 62 para reconstruir el bloque codificado para su uso como una imagen de referencia.
- La unidad de predicción intra 46 dentro de la unidad de predicción 41 puede llevar a cabo la codificación intra-predictiva del bloque de vídeo actual en relación con uno o más bloques vecinos en el mismo marco o porción como el bloque actual a codificar para proporcionar una compresión espacial. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 dentro de la unidad de predicción 41 realizan codificación inter-predictiva del bloque de vídeo actual en relación con uno o más bloques de predicción en una o más imágenes de referencia para proporcionar una compresión temporal.
- La unidad de estimación de movimiento 42 puede estar configurada para determinar el modo de inter-predicción para una porción de vídeo de acuerdo con un patrón predeterminado para una secuencia de vídeo. El patrón predeterminado puede designar porciones de vídeo en la secuencia como porciones P, porciones B o porciones GPB (generalizada P/B). La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden estar muy integradas, pero se ilustran por separado para fines conceptuales. La estimación de movimiento, realizada por la unidad de estimación de movimiento 42, es el proceso de generación de vectores de movimiento, que estiman el movimiento de los bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de una PU de un bloque de vídeo dentro de un marco actual de vídeo o una imagen con relación a un bloque de predicción dentro de una imagen de referencia.
- Un bloque de predicción es un bloque que se encuentra que se asemeja mucho a la PU del bloque de vídeo a codificar en términos de diferencia de píxeles, que puede determinarse por la suma de la diferencia absoluta (SAD), la suma de la diferencia cuadrada (SSD), u otras métricas de diferencia. En algunos ejemplos, el codificador 20 de vídeo puede calcular los valores de las posiciones de píxeles sub-enteros de imágenes de referencia almacenadas en la memoria de imágenes de referencia 64. Por ejemplo, el codificador 20 de vídeo puede interpolar valores de las posiciones de un cuarto de píxel, las posiciones de un octavo de píxel, u otras posiciones de píxel fraccionarias de la imagen de referencia. Por lo tanto, la unidad de estimación de movimiento 42 puede realizar una búsqueda de movimiento en relación con las posiciones de píxel completas y las posiciones de píxel fraccionadas y emitir un vector de movimiento con una precisión de píxel fraccionario.
- La unidad de estimación de movimiento 42 calcula un vector de movimiento para una PU de un bloque de vídeo en una porción inter-codificada mediante la comparación de la posición de la PU con la posición de un bloque de predicción de una imagen de referencia. La imagen de referencia se puede seleccionar de una primera lista de imágenes de referencia (Lista 0) o una segunda lista de imágenes de referencia (Lista 1), cada una de las cuales identifica una o más imágenes de referencia almacenadas en la memoria de imágenes de referencia 64. La unidad de estimación de movimiento 42 envía el vector de movimiento calculado a la unidad de codificación de entropía 56 y a la unidad de compensación de movimiento 44.
- La compensación de movimiento, realizada por la unidad de compensación de movimiento 44, puede implicar ir a buscar o generar el bloque de predicción basado en el vector de movimiento determinado por la estimación de movimiento, posiblemente realizando interpolaciones con precisión de sub-píxel. Al recibir el vector de movimiento de la PU del bloque de vídeo actual, la unidad de compensación de movimiento 44 puede localizar el bloque predictivo al que apunta el vector de movimiento en una de las listas de imágenes de referencia. El codificador 20 de vídeo forma un bloque de vídeo residual restando los valores de píxel del bloque de predicción de los valores de píxel del bloque de vídeo actual que se está codificando, formando valores de diferencia de píxel. Los valores de diferencia de píxel forman datos residuales para el bloque, y pueden incluir componentes de diferencia de luminancia y croma. El sumador 50 representa el componente o componentes que realizan esta operación de resta. La unidad de compensación de movimiento 44 también puede generar elementos sintácticos asociados con los bloques de vídeo y la porción de vídeo para su uso por parte del decodificador 30 de vídeo en la decodificación de los bloques de vídeo de la porción de vídeo.

La unidad de intra-predicción 46 puede intra-predecir un bloque actual, como una alternativa a la inter-predicción realizada por la unidad de compensación de movimiento 42 y la unidad de estimación de movimiento 44, tal como se describe anteriormente. En particular, la unidad de intra-predicción 46 puede determinar un modo de intra-predicción que se usará para codificar un bloque actual. En algunos ejemplos, la unidad de intra-predicción 46 puede codificar un bloque actual usando varios modos de intra-predicción, por ejemplo, durante pasadas de codificación separadas, la unidad de intra-predicción 46 (o unidad de selección de modo 40, en algunos ejemplos) puede seleccionar un modo de intra-predicción apropiado a utilizar a partir de los modos probados. Por ejemplo, la unidad de intra-predicción 46 puede calcular los valores de tasa de distorsión utilizando un análisis de tasa de distorsión para los distintos modos de intra-predicción probados, y seleccionar el modo de intra-predicción que tiene las mejores características de la tasa de distorsión entre los modos probados. El análisis de la tasa de distorsión generalmente determina una cantidad de distorsión (o errores) entre un bloque codificado y un bloque original sin codificar, que fue codificado para producir el bloque codificado, así como una tasa de bits (es decir, un número de bits) utilizado para producir el bloque codificado. La unidad de intra-predicción 46 puede calcular los coeficientes de las distorsiones y las tasas de los distintos bloques codificados para determinar qué modo de intra-predicción exhibe el mejor valor de tasa de distorsión para el bloque.

En cualquier caso, después de seleccionar un modo de intra-predicción para un bloque, la unidad de intra-predicción 46 puede proporcionar información indicativa del modo de intra-predicción seleccionado para el bloque a la unidad de codificación de entropía 56. La unidad de codificación de entropía 56 puede codificar la información que indica el modo de intra-predicción seleccionado de acuerdo con las técnicas de esta divulgación. El codificador 20 de vídeo puede incluir en los datos de configuración el flujo de bits de transmisión, que puede incluir una pluralidad de tablas de índice del modo de intra-predicción y una pluralidad de tablas de índice de modo modificadas de intra-predicción (también denominadas tablas de asignación de palabras de código), definiciones de contextos de codificación para varios bloques, e indicaciones de un modo más probable de intra-predicción, una tabla de índice de modo intra-predicción, y una tabla de índice de modo intra-predicción modificada a utilizar para cada uno de los contextos.

Después la unidad de predicción 41 genera el bloque de predicción para el bloque de vídeo actual, ya sea a través de inter-predicción o intra-predicción, el codificador 20 de vídeo forma un bloque de vídeo residual restando el bloque de predicción del bloque de vídeo actual. Los datos de vídeo residuales en el bloque residual se pueden incluir en una o más unidades de formación y se aplican a la unidad de transformación 52. La unidad de transformación 52 transforma los datos de vídeo residuales en coeficientes de transformación residuales utilizando una transformación, tal como una transformada de coseno discreta (DCT) o una transformación conceptualmente similar. La unidad de transformación 52 puede convertir los datos de vídeo residuales de un dominio de píxel a un dominio de la transformada, tal como un dominio de frecuencia.

La unidad de transformación 52 podrá enviar los coeficientes resultantes de transformación a la unidad de cuantificación 54. La unidad de cuantificación 54 cuantifica los coeficientes de transformación para reducir aún más la velocidad de bits. El proceso de cuantificación puede reducir la profundidad de bits asociada con algunos o todos los coeficientes. El grado de cuantificación puede modificarse mediante el ajuste de un parámetro de cuantificación. En algunos ejemplos, la unidad de cuantificación 54 puede entonces realizar una exploración de la matriz incluyendo los coeficientes de transformación cuantificados. Alternativamente, la unidad de codificación de entropía 56 puede realizar la exploración. Como un ejemplo, las técnicas de codificación descritas en esta divulgación se pueden llevar a cabo total o parcialmente mediante la unidad de codificación de entropía 56. Sin embargo, los aspectos de esta divulgación no son tan limitados. Por ejemplo, las técnicas de codificación descritas en esta divulgación pueden realizarse mediante un componente del codificador 20 de vídeo que no se muestra en la figura 2, tal como un procesador o cualquier otro componente. En algunos ejemplos, las técnicas de codificación de esta divulgación pueden realizarse mediante una de las otras unidades o módulos ilustrados en la figura 2. En todavía otros ejemplos, las técnicas de codificación de esta divulgación pueden realizarse mediante una fusión de unidades y módulos del codificador 20 de vídeo. De esta manera, el codificador 20 de vídeo puede estar configurado para realizar las técnicas de ejemplo descritas en esta divulgación.

Después de la cuantificación, la entropía en la unidad de codificación de entropía 56 codifica los coeficientes de transformación cuantificados. Por ejemplo, la unidad de codificación de entropía 56 puede llevar a cabo la codificación de longitud variable adaptativa según el contexto (CAVLC), la codificación aritmética binaria adaptativa según el contexto (CABAC), basada en la codificación aritmética binaria adaptativa según el contexto basada en la sintaxis (SBAC), la codificación de entropía de partición del intervalo de probabilidad (PIPE) u otra metodología o técnica de codificación de entropía. Después de la codificación de entropía mediante la unidad de codificación de entropía 56, el flujo de bits codificado puede ser transmitido al decodificador 30 de vídeo, o archivarse para su posterior transmisión o recuperación por parte del decodificador de vídeo 30. La unidad de codificación de entropía 56 también puede codificar la entropía de los vectores de movimiento y los demás elementos de sintaxis para la porción de vídeo actual que se está codificando.

En un ejemplo de la divulgación, la unidad de codificación de entropía 56 puede estar configurada para determinar un primer tipo de predicción para un bloque de datos de vídeo en una porción P, que representa el primer tipo de predicción como un elemento de sintaxis del tipo de predicción de porción P, determinar una segunda predicción escribir para un bloque de datos de vídeo en una porción B, representar el segundo tipo de predicción como un elemento de sintaxis de tipo de predicción de porción B, determinar una binarización de porción P para el elemento

de sintaxis del tipo de predicción de porción P, determinar una binarización de porción B para el elemento de sintaxis del tipo de predicción de porción B, en el que el elemento de sintaxis del tipo de predicción de porción P y el elemento de sintaxis del tipo de predicción de porción B se determinan usando la misma lógica de binarización y codificar los datos de vídeo en base a los binarizaciones del elemento de sintaxis del tipo de predicción de porción P y el elemento de sintaxis de predicción de porción B.

En otro ejemplo de la divulgación, la unidad de codificación de entropía 56 puede estar configurada para determinar un tipo de partición para un modo de predicción para un bloque de datos de vídeo, codificar un bin de tipo de partición de un elemento de sintaxis del tipo de predicción para el bloque de datos de vídeo usando codificación aritmética binaria adaptativa según el contexto con un solo contexto, en el que el contexto individual es el mismo para cualquier tipo de partición, y codifican un bin de tamaño de la partición de un elemento de sintaxis del tipo de predicción para el bloque de datos de vídeo usando la codificación aritmética binaria adaptativa al contexto en el modo de derivación.

En otro ejemplo de la divulgación, la unidad de codificación de entropía 56 puede estar configurada para codificar un indicador de bloque codificado Cb croma para un bloque de datos de vídeo usando la codificación aritmética binaria adaptativa según el contexto (CABAC), en la que CABAC utiliza un conjunto de contexto que incluye uno o más contextos, y el código de una señal bloque croma codificada Cr utilizando CABAC, en CABAC utiliza el mismo conjunto de contexto que el indicador de bloque Cb croma codificado. El codificador 20 de vídeo y el decodificador 30 de vídeo pueden estar configurados además para seleccionar un contexto a partir de los uno o más contextos basados en 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 transformación inversa 60 aplican cuantificación inversa y transformación inversa, respectivamente, para reconstruir el bloque residual en el dominio de píxel para su uso posterior como un bloque de referencia de una imagen de referencia. La unidad de compensación de movimiento 44 puede calcular un bloque de referencia añadiendo el bloque residual a un bloque de predicción de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. La unidad de compensación de movimiento 44 también puede aplicar uno o más filtros de interpolación al bloque residual reconstruido para calcular los valores de píxel sub-enteros para su uso en la estimación de movimiento. El sumador 62 añade al bloque residual reconstruido el bloque de predicción con compensación de movimiento producido por unidad de compensación de movimiento 44 para producir un bloque de referencia para su almacenamiento en la memoria de imágenes de referencia 64. El bloque de referencia puede ser utilizado por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 como un bloque de referencia para inter-predicir un bloque en un marco de vídeo o imagen posterior.

La figura 3 es un diagrama de bloques que ilustra un decodificador 30 de vídeo de ejemplo que pueden aplicar las técnicas descritas en esta divulgación. En el ejemplo de la figura 3, el decodificador 30 de vídeo incluye una unidad de decodificación de 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 de movimiento 82 y la unidad de intra-predicción 84. El decodificador 30 de vídeo puede, en algunos ejemplos, llevar a cabo una pasada de decodificación generalmente recíproca a la pasada de codificación descrita con respecto al codificador 20 de vídeo de la figura 2.

Durante el proceso de decodificación, el decodificador 30 de vídeo recibe un flujo de bits de vídeo codificado que representa los bloques de vídeo de una sección de vídeo codificada y elementos sintácticos asociados del codificador 20 de vídeo. La unidad de decodificación de entropía 80 del decodificador 30 de vídeo decodifica el flujo de bits para generar coeficientes cuantificados, vectores de movimiento y otros elementos de sintaxis. La unidad de decodificación de 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 en el nivel de porción de vídeo y/o el nivel de bloque de vídeo.

Como un ejemplo, las técnicas de codificación descritas en esta divulgación se pueden llevar a cabo total o parcialmente por la unidad de decodificación de entropía 80. Sin embargo, los aspectos de esta divulgación no son tan limitados. Por ejemplo, las técnicas de codificación descritas en esta divulgación pueden realizarse mediante un componente del decodificador 30 de vídeo que no se muestra en la figura 3, tal como un procesador o cualquier otro componente. En algunos ejemplos, las técnicas de codificación de esta divulgación pueden realizarse mediante una de las otras unidades o módulos ilustrados en la figura 3. En todavía otros ejemplos, las técnicas de codificación de esta divulgación pueden realizarse mediante una fusión de unidades y módulos del decodificador 30 de vídeo. De esta manera, el decodificador 30 de vídeo puede estar configurado para realizar las técnicas de ejemplo descritas en esta divulgación.

En un ejemplo de la divulgación, la unidad de decodificación de entropía 80 puede estar configurada para asignar un elemento de sintaxis de tipo de predicción de porción P binarizado a un tipo de predicción usando una asignación de binarización para un bloque de datos de vídeo en una porción P, asignar una elemento de sintaxis de tipo de predicción de porción B binarizado a un tipo de predicción utilizando la misma asignación de binarización para un bloque de datos de vídeo en una porción B, y decodificar los datos de vídeo en base a los tipos de predicción

asignados.

En un ejemplo de la divulgación, la unidad de decodificación de entropía 80 puede estar configurada para recibir un elemento de sintaxis de tipo de predicción para un bloque de datos de vídeo que ha sido codificado usando la codificación aritmética binaria adaptativa según el contexto (CABAC), el elemento de sintaxis de tipo de predicción que incluye un tipo de partición bin que representa un tipo de partición y un bin de tamaño de la partición que representa un tamaño de partición, decodificar el bin de tipo de partición del elemento de sintaxis de tipo de predicción usando el contexto de la aritmética binaria adaptativa de codificación con un solo contexto, en el que el contexto individual es el mismo para cualquier tipo de partición, y decodificar el tamaño de partición bin de la sintaxis de tipo de predicción para un bloque de datos de vídeo utilizando codificación aritmética binaria adaptativa según el contexto en el modo de derivación.

En otro ejemplo de la divulgación, la unidad de decodificación de entropía 80 puede estar configurada para codificar un indicador de bloque codificado Cb croma para un bloque de datos de vídeo usando la codificación aritmética binaria adaptativa según el contexto (CABAC), en el que CABAC utiliza un conjunto de contexto que incluye uno o más contextos, y el código de una señal bloque croma codificada Cr utiliza CABAC, en el que CABAC utiliza el mismo conjunto de contexto como el indicador de bloque Cb croma codificado. El codificador 20 de vídeo y el decodificador 30 de vídeo pueden estar configurados además para seleccionar un contexto a partir de los uno o más contextos basados en una profundidad de transformación de una unidad de transformación asociada con el bloque de datos de vídeo.

Cuando la porción de vídeo se codifica como porción intra-codificada (I), la unidad de intra-predicción 84 de la unidad de predicción 81 puede generar datos de predicción para un bloque de vídeo de la porción de vídeo actual, basado en un modo de intra-predicción señalado y los datos de los bloques previamente decodificados del marco o imagen actual. Cuando el marco de vídeo se codifica como una porción inter-codificada, la unidad de compensación de movimiento 82 de la unidad de predicción 81 produce bloques de predicción para un bloque de vídeo de la porción de vídeo actual basado en los vectores de movimiento y otros elementos de sintaxis (es decir, B, P o GPB) recibidos desde la unidad de decodificación de entropía 80. Los bloques de predicción se pueden producir a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El decodificador 30 de vídeo puede construir las listas de marco de referencia, la lista 0 y la lista 1, utilizando técnicas de construcción predeterminadas basadas en imágenes de referencia almacenadas en la memoria de imágenes de referencia 92.

La unidad de compensación de movimiento 82 determina la información de predicción para un bloque de vídeo de la sección de vídeo actual analizando los vectores de movimiento y otros elementos de sintaxis, y utiliza la información de predicción para producir los bloques de predicción para el bloque de vídeo actual que está siendo decodificado. Por ejemplo, la unidad de compensación de movimiento 82 utiliza algunos de los elementos de sintaxis recibidos para determinar un modo de predicción (por ejemplo, intra o inter-predicción) utilizado para codificar los bloques de vídeo de la sección de vídeo, un tipo de sector de inter-predicción (por ejemplo, porción B, porción P, o una porción GPB), la información de construcción de una o más de las listas de imágenes de referencia para la división, los vectores de movimiento para cada bloque de vídeo inter-codificada de la condición de la porción, inter-predicción para cada bloque de vídeo inter-codificada de la porción, y otra información para decodificar los bloques de vídeo en la sección de vídeo actual.

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

La unidad de cuantificación inversa 86 cuantifica inversas, es decir, de-cuantifica, los coeficientes de transformación cuantificados previstos en el flujo de bits y decodificados por la unidad de decodificación de entropía 80. El proceso de cuantificación inversa puede incluir el uso de un parámetro de cuantificación calculado por el codificador de vídeo 20 para cada bloque de vídeo en la porción de vídeo para determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa que se debe aplicar. La unidad de transformación inversa 88 aplica una transformada inversa, por ejemplo, una DCT inversa, una transformada inversa de número entero, o una inversa conceptualmente similar al proceso de transformación, los coeficientes de transformación para producir bloques residuales en el dominio de píxeles.

Después de que la unidad de compensación de movimiento 82 genera el bloque de predicción para el bloque de vídeo actual basado en los vectores de movimiento y otros elementos de sintaxis, el decodificador 30 de vídeo forma un bloque de vídeo decodificado sumando los bloques residuales de la unidad de transformación inversa 88 con los bloques de predicción correspondientes generados por la unidad de compensación de movimiento 82. El sumador 90 representa el componente o componentes que realizan esta operación de suma. Si se desea, un filtro de desbloqueo también puede aplicarse para filtrar los bloques decodificados para eliminar artefactos de formación de los bloques. Otros filtros de bucle (ya sea en el bucle de codificación o después del bucle de codificación) también pueden ser utilizados para suavizar las transiciones de píxeles, o mejorar de otra manera la calidad del vídeo. Los

bloques de vídeo decodificados en un marco o cuadro dado se almacenan en la memoria de imágenes de referencia 92, que almacena imágenes de referencia utilizadas para la compensación de movimiento posterior. La memoria de imágenes de referencia 92 también almacena vídeo decodificado para su presentación posterior en un dispositivo de visualización, como el dispositivo de visualización 32 de la figura 1.

- 5 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 20 de vídeo. El codificador 20 de vídeo puede estar configurado para determinar un primer tipo de predicción para un bloque de datos de vídeo en una porción P (602), y para representar el primer tipo de predicción como un elemento de sintaxis de tipo de predicción de porción P (604). El codificador 20 de vídeo puede estar configurado además para determinar un
10 segundo tipo de predicción para un bloque de datos de vídeo en una porción B (606), y para representar el segundo tipo de predicción como un elemento de sintaxis de tipo de predicción de porción B (608). El elemento de sintaxis de tipo de predicción de porción P y el elemento de sintaxis de tipo de predicción de porción B especifican un modo de predicción y un tipo de partición. El modo de predicción puede incluir uno de entre la predicción y la intra-predicción. El tipo de partición puede incluir una de las particiones simétricas y particiones asimétricas.
- 15 El codificador 20 de vídeo puede estar configurado además para determinar una binarización de porción P para el elemento de sintaxis de tipo de predicción de porción P (610), y para determinar una binarización de porción B para el elemento de sintaxis de tipo de predicción de porción B, en el que el elemento de sintaxis de tipo de predicción de porción P y el elemento de sintaxis de tipo de predicción de porción B se determinan usando la misma lógica de binarización (612). El codificador de vídeo 20 puede entonces codificar los datos de vídeo basado en las
20 binarizaciones del elemento de sintaxis de tipo de predicción de porción P y el elemento de sintaxis de tipo de predicción de porción B (614).

La codificación de los datos de vídeo puede comprender binarizar el elemento de sintaxis de tipo de predicción de porción P con las binarizaciones de porción P determinadas, binarizar el elemento de sintaxis de tipo de predicción de porción B con las binarizaciones de porción B decididas, aplicar la codificación aritmética binaria adaptativa según el contexto (CABAC) al elemento de sintaxis de tipo de predicción de porción P binarizado, y la codificación aritmética binaria adaptativa según el contexto (CABAC) solicitada al elemento de sintaxis de tipo de predicción de
25 porción 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 30 de vídeo. El decodificador 30 de vídeo puede estar configurado para recibir un elemento de sintaxis de tipo de predicción de porción P de la codificación aritmética binaria adaptativa según el contexto que indica el tipo de predicción para el
30 bloque de datos de vídeo en una porción P (702), y para recibir un elemento de sintaxis de tipo de predicción de porción B de codificación aritmética binaria adaptativa según el contexto que indica el tipo de predicción para el bloque de datos de vídeo en una porción B (704). El elemento de sintaxis de tipo de predicción de porción P y el
35 elemento de sintaxis de tipo de predicción de porción B especifican un modo de predicción y un tipo de partición. El modo de predicción puede incluir una de entre la predicción y la intra-predicción. El tipo de partición puede incluir una de las particiones simétricas y particiones asimétricas.

El decodificador de vídeo 30 puede estar configurado además para decodificar el elemento sintaxis de tipo de predicción de porción P para producir un elemento de sintaxis de tipo de predicción de porción P binarizado (706), y para decodificar el elemento de sintaxis de tipo de predicción de porción B para producir un elemento de sintaxis de tipo de predicción de porción B (708) binarizado. El decodificador 30 de vídeo puede estar configurado además para asignar el elemento de sintaxis de tipo de predicción de porción P binarizado a un tipo de predicción usando una asignación de binarización para un bloque de datos de vídeo en una porción P (710), y para asignar el elemento de
40 sintaxis de tipo de predicción de porción B binarizado a un tipo de predicción utilizando la misma asignación de binarización para un bloque de datos de vídeo en una porción B (712). El decodificador 30 de vídeo puede entonces descifrar los datos de vídeo en base a los tipos de predicción asignados (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 20 de vídeo puede estar configurado para determinar un tipo de partición para un modo de predicción para un
50 bloque de datos de vídeo (802) y para codificar un marco de un elemento de sintaxis de tipo de partición del tipo de predicción para el bloque de datos de vídeo utilizando codificación aritmética binaria adaptativa según el contexto (CABAC) con un solo 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 bin de tipo de partición indica si la partición asimétrica se divide verticalmente o se divide horizontalmente. Por ejemplo, el tamaño de la partición bin indica si una primera partición es una cuarta parte de un tamaño del bloque de datos de vídeo o si la primera partición es tres cuartas partes de la
55 dimensión del bloque de datos de vídeo.

El codificador 20 de vídeo puede estar configurado además para codificar un bin de tamaño de la partición del elemento de sintaxis de tipo de predicción para el bloque de datos de vídeo utilizando CABAC en el modo de derivación (806).

La figura 9 es un diagrama de flujo que ilustra un procedimiento de decodificación vídeo de ejemplo de la divulgación. El procedimiento de la figura 9 puede implementarse mediante el decodificador 30 de vídeo. El decodificador 30 de vídeo puede estar configurado para recibir un elemento de sintaxis de tipo de predicción para un bloque de datos de vídeo que se ha codificado utilizando codificación aritmética binaria adaptativa según el contexto (CABAC), incluyendo el elemento de sintaxis de tipo de predicción una partición de tipo bin que representa un tipo de partición y un bin de tamaño de la partición que representa un tamaño de la partición (902). En un ejemplo, el tipo de partición es una partición asimétrica y el bin de tipo de partición indica si la partición asimétrica se divide verticalmente o se divide horizontalmente. Por ejemplo, el tamaño de la partición bin indica si una primera partición es una cuarta parte de un tamaño del bloque de datos de vídeo o si la primera partición es tres cuartas partes de la dimensión del bloque de datos de vídeo.

El decodificador 30 de vídeo puede estar configurado además para decodificar el bin de tipo de partición del elemento de sintaxis de tipo de predicción utilizando CABAC con un solo contexto, en el que el contexto individual es el mismo para cualquier tipo de partición (904), y para decodificar el tamaño de partición bin del elemento de sintaxis tipo de predicción usando CABAC en el modo de derivación (906).

La figura 10 es un diagrama de flujo que ilustra un procedimiento de codificación vídeo de ejemplo de la divulgación. El procedimiento de la figura 10 puede implementarse mediante cualquiera del codificador 20 de vídeo o el decodificador de vídeo. Para los fines de la figura 10, al codificador 20 de vídeo y al decodificador 30 de vídeo se hará referencia a, colectivamente, como 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 croma codificado Cb para un bloque de datos de vídeo usando la codificación aritmética binaria adaptativa según el contexto (CABAC), en el que la codificación del indicador de bloque croma codificado Cb comprende la utilización de un conjunto de contexto que incluye uno o más contextos como parte de la CABAC (1002), y para codificar un indicador de bloque croma codificado Cr usando CABAC, en el que la codificación del indicador de bloque croma codificado Cr comprende utilizar el mismo conjunto de contexto que el indicador de bloque croma codificado 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 estar configurado además para seleccionar un contexto a partir de los uno o más contextos basados en 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 estar configurado además para indicar el indicador de bloque croma Cb codificado que se codifica en un flujo de bits de vídeo codificado, y para señalar el indicador de bloque croma codificado Cr que se codifica en el flujo de bits de vídeo codificado. Cuando funciona como un decodificador de vídeo, un codificador de vídeo puede estar configurado además para recibir el indicador de bloque croma codificado Cb que se codifica en un flujo de bits de vídeo codificado, y para recibir el indicador de bloque croma codificado Cr que se codifica en el flujo de bits de vídeo codificado.

En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, firmware, o cualquier combinación de los mismos. Si se implementan en software, las funciones pueden almacenarse o transmitirse a través de un medio legible por ordenador, como una o más instrucciones o código, y ejecutarse mediante una unidad de procesamiento basado en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legible por ordenador, lo que corresponde a un medio tangible tal como un medio de almacenamiento de datos, o medios de comunicación incluyendo cualquier medio que facilite la transferencia de un programa informático de un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador generalmente pueden corresponder a (1) medios de almacenamiento tangibles legibles por ordenador que no son transitorios o (2) un medio de comunicación tal como una onda portadora o señal. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que se puede acceder mediante uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la aplicación de las técnicas descritas en esta divulgación. Un producto de programa de ordenador puede incluir un medio legible por ordenador.

A modo de ejemplo, y no de limitación, tales medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento en disco óptico, almacenamiento en disco magnético u otros dispositivos de almacenamiento magnético, memoria flash, o cualquier otro medio que se pueda utilizar para almacenar el código del programa deseado en forma de instrucciones o estructuras de datos y que se puede acceder mediante un ordenador. Además, cualquier conexión se denomina correctamente un medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten de una página web, servidor u otra fuente remota mediante un cable coaxial, cable de fibra óptica, par trenzado, línea de abonado digital (DSL), o tecnologías inalámbricas como infrarrojos, radio y microondas, entonces el cable coaxial, cable de fibra óptica, par trenzado, DSL o tecnologías inalámbricas como infrarrojos, radio y microondas están incluidos en la definición de los medios. Debe entenderse, sin embargo, que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales, u otros medios transitorios, sino que se dirigen a medios de almacenamiento tangibles no transitorios. Disco, como se usa aquí, incluye disco compacto (CD), discos láser, discos ópticos, discos versátiles digitales (DVD), y discos Blu-ray, donde los discos generalmente reproducen datos magnéticamente, mientras que los discos reproducen los datos ópticamente con láser.

Combinaciones de los anteriores también deben incluirse dentro del alcance de medios legibles por ordenador.

5 Las instrucciones pueden ser ejecutadas por uno o más procesadores, tales como uno o más procesadores digitales de señal (DSP), microprocesadores de propósito general, circuitos integrados de aplicación específica (ASIC), matrices de campo lógicas programables (FPGA), u otra circuitería equivalente integrada o lógica discreta. Por
10 consiguiente, el término "procesador", como se usa en el presente documento puede referirse a cualquiera de la estructura anterior o cualquier otra estructura adecuada para la aplicación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en este documento puede proporcionarse dentro de módulos de hardware y/o software dedicados configurados para codificar y decodificar, o incorporada en un códec combinado. Además, las técnicas podrían aplicarse plenamente en uno o más circuitos o elementos lógicos.

15 Las técnicas de esta divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluyendo un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de circuitos integrados (por ejemplo, un conjunto de chips). Varios componentes, módulos o unidades se describen en esta divulgación para enfatizar los aspectos funcionales de los dispositivos configurados para realizar las técnicas descritas, pero no necesariamente requieren la realización de diferentes unidades de hardware. Más bien, como se ha descrito anteriormente, diversas unidades pueden combinarse en una unidad de codificación de hardware o proporcionarse mediante una colección de unidades de hardware interoperativas, incluyendo uno o más procesadores, como se describió anteriormente, en conjunción con el software y/o firmware adecuado.

20 Se han descrito Varios ejemplos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.

REIVINDICACIONES

1. Un procedimiento de codificación de datos de vídeo que comprende:

5 seleccionar (1006) un contexto de uno o más contextos basados en una profundidad de transformación de una unidad de transformación asociada con un bloque de datos de vídeo a codificar, en el que la profundidad de transformación especifica el nivel de subdivisión actual de una unidad de codificación en bloques para el objeto de codificación de transformación;

10 codificar (1002) un indicador de bloque codificado croma Cb para el bloque de datos de vídeo usando la codificación aritmética binaria adaptativa al contexto, CABAC y el contexto seleccionado, en el que la codificación del indicador de bloque codificado croma Cb comprende el uso de un conjunto de contexto que incluye uno o más contextos; y

codificar (1004) un indicador de bloque codificado croma Cr usando CABAC, en el que la codificación del indicador de bloque codificado croma Cr comprende utilizar el mismo conjunto de contexto como el indicador de bloque codificado croma Cb.

2. Un procedimiento de decodificación de datos de vídeo que comprende:

15 seleccionar (1006) un contexto de uno o más contextos basados en una profundidad de transformación de una unidad de transformación asociada con un bloque de datos de vídeo a codificar, en el que la profundidad de transformación especifica el nivel de subdivisión actual de una unidad de codificación en bloques para el objeto de codificación de transformación;

20 decodificar (1002) un indicador de bloque codificado croma Cb para el bloque de datos de vídeo usando la codificación aritmética binaria adaptativa al contexto, CABAC y el contexto seleccionado, en el que la decodificación del indicador de bloque codificado croma Cb comprende el uso de un conjunto de contexto que incluye el uno o más contextos; y

25 decodificar (1004) un indicador de bloque codificado croma Cr usando CABAC, en el que la decodificación del indicador de bloque codificado croma Cr comprende utilizar el mismo conjunto de contexto que el indicador de bloque codificado croma Cb.

3. El procedimiento de la reivindicación 1 ó 2, en el que el conjunto de contexto incluye 5 contextos.

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

señalar el indicador de bloque codificado croma Cb en un flujo de bits de vídeo codificado; y
señalar el indicador de bloque codificado croma Cr en el flujo de bits de vídeo codificado.

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

recibir el indicador de bloque codificado croma Cb en un flujo de bits de vídeo codificado; y
recibir el indicador de bloque codificado croma Cr en el flujo de bits de vídeo codificado.

6. Un aparato (20) configurado para codificar datos de vídeo que comprende:

35 medios (56) para seleccionar un contexto de uno o más contextos basados en una profundidad de transformación de una unidad de transformación asociada con un bloque de datos de vídeo a codificar, en el que la profundidad de transformación especifica el nivel de subdivisión actual de una unidad de codificación en bloques para el objeto de codificación de transformación;

40 medios (56) para la codificación de un indicador de bloque codificado croma Cb para el bloque de datos de vídeo usando la codificación aritmética binaria adaptativa al contexto, CABAC, y el contexto seleccionado, en el que la codificación del indicador de bloque codificado croma Cb comprende el uso de un conjunto contexto que incluye uno o más contextos; y

medios (56) para la codificación de un indicador de bloque codificado croma Cr usando CABAC, en el que la codificación del indicador de bloque codificado croma Cr comprende utilizar el mismo conjunto de contexto que el indicador de bloque codificado croma Cb.

45 7. Un aparato (30) configurado para decodificar datos de vídeo que comprende:

medios (80) para seleccionar un contexto de uno o más contextos basados en una profundidad de transformación de una unidad de transformación asociada con un bloque de datos de vídeo a codificar, en el que la profundidad de transformación especifica el nivel de subdivisión actual de una unidad de codificación en bloques para el objeto de codificación de transformación;

50 medios (80) para decodificar un indicador de bloque codificado croma Cb para el bloque de datos de vídeo usando la codificación aritmética binaria adaptativa al contexto, CABAC, y el contexto seleccionado, en el que la decodificación del indicador de bloque codificado croma Cb comprende el uso de un conjunto de contexto que incluye uno o más contextos; y

55 medios (80) para decodificar un indicador de bloque codificado croma Cr usando CABAC, en el que la decodificación del indicador de bloque codificado croma Cr comprende utilizar el mismo conjunto de contexto

que el indicador de bloque codificado croma Cb.

8. El aparato de la reivindicación 6 ó 7, en el que el conjunto de contexto incluye 5 contextos.

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

5 medios para la señalización del indicador de bloque codificado croma Cb en un flujo de bits de vídeo codificado;
 y
 medios para la señalización del indicador de bloque codificado croma Cr en el flujo de bits de vídeo codificado.

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

 medios para recibir el indicador de bloque codificado croma Cb en un flujo de bits de vídeo codificado; y
 medios para recibir el indicador de bloque codificado croma Cr en el flujo de bits de vídeo codificado.

10 11. Un medio de almacenamiento legible por ordenador que almacena instrucciones que, al ejecutarse, originan que uno o más procesadores configurados para codificar o decodificar datos de vídeo efectúen el procedimiento de cualquiera de las reivindicaciones 1 a 5.

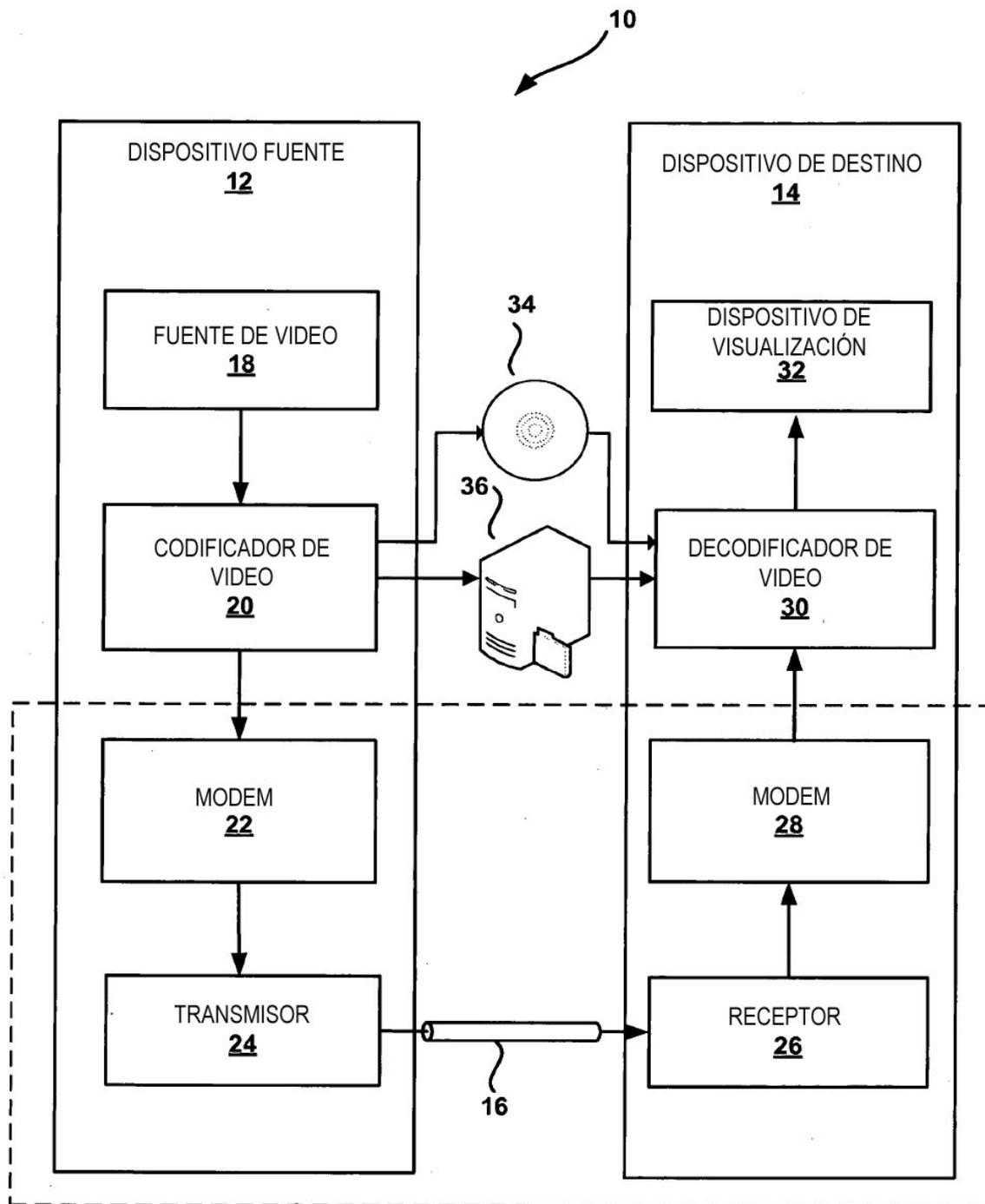


FIG. 1

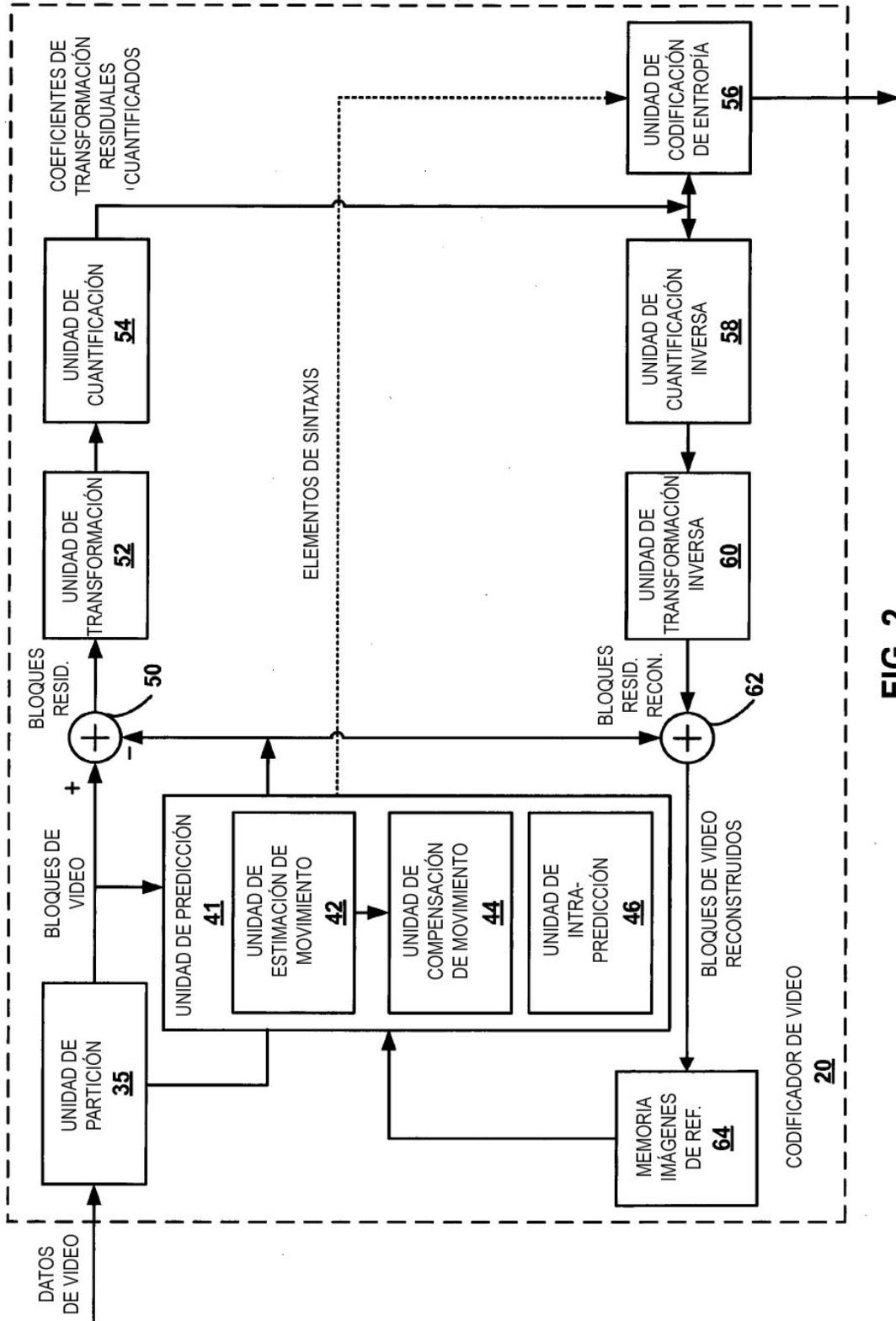


FIG. 2

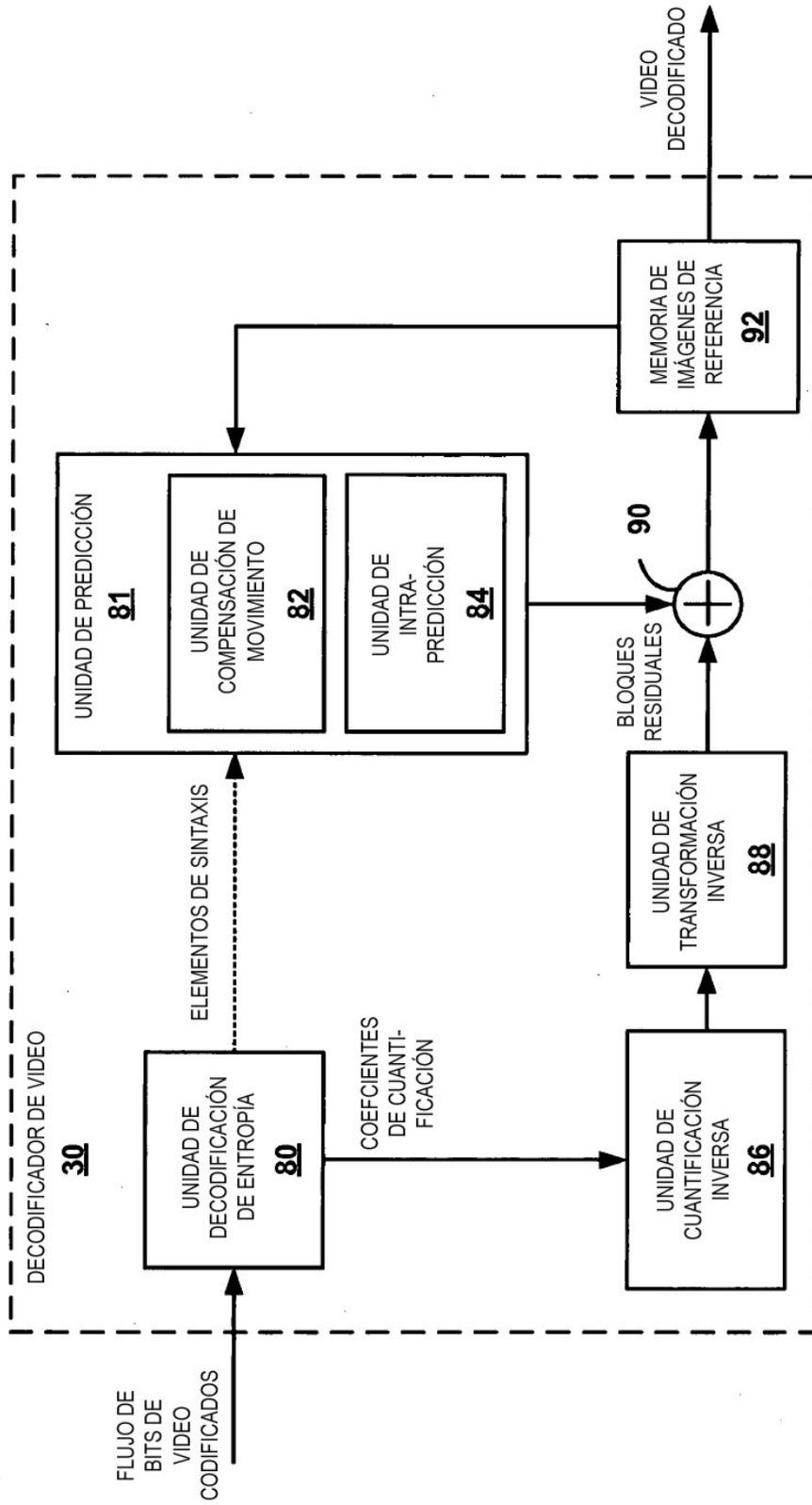


FIG. 3

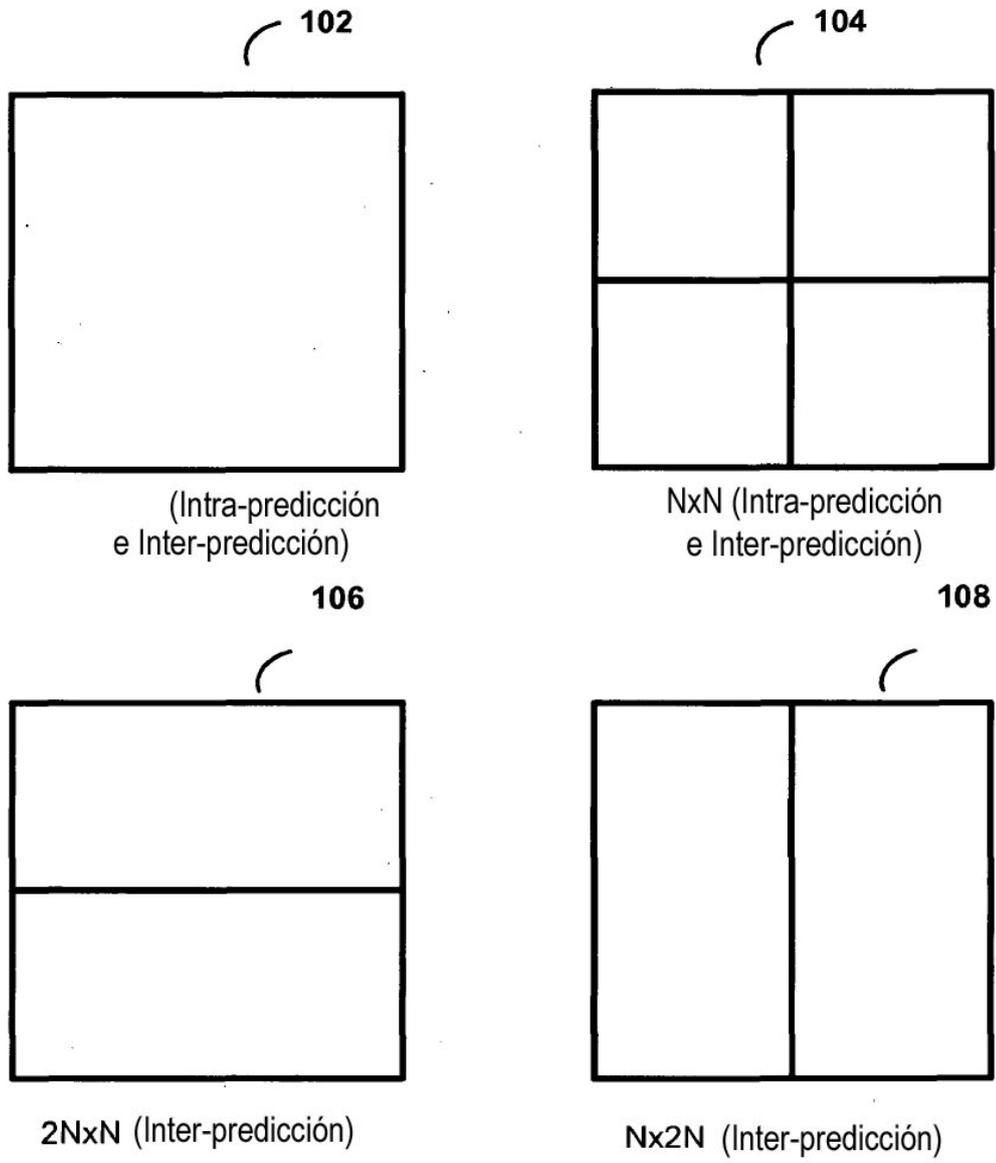


FIG. 4

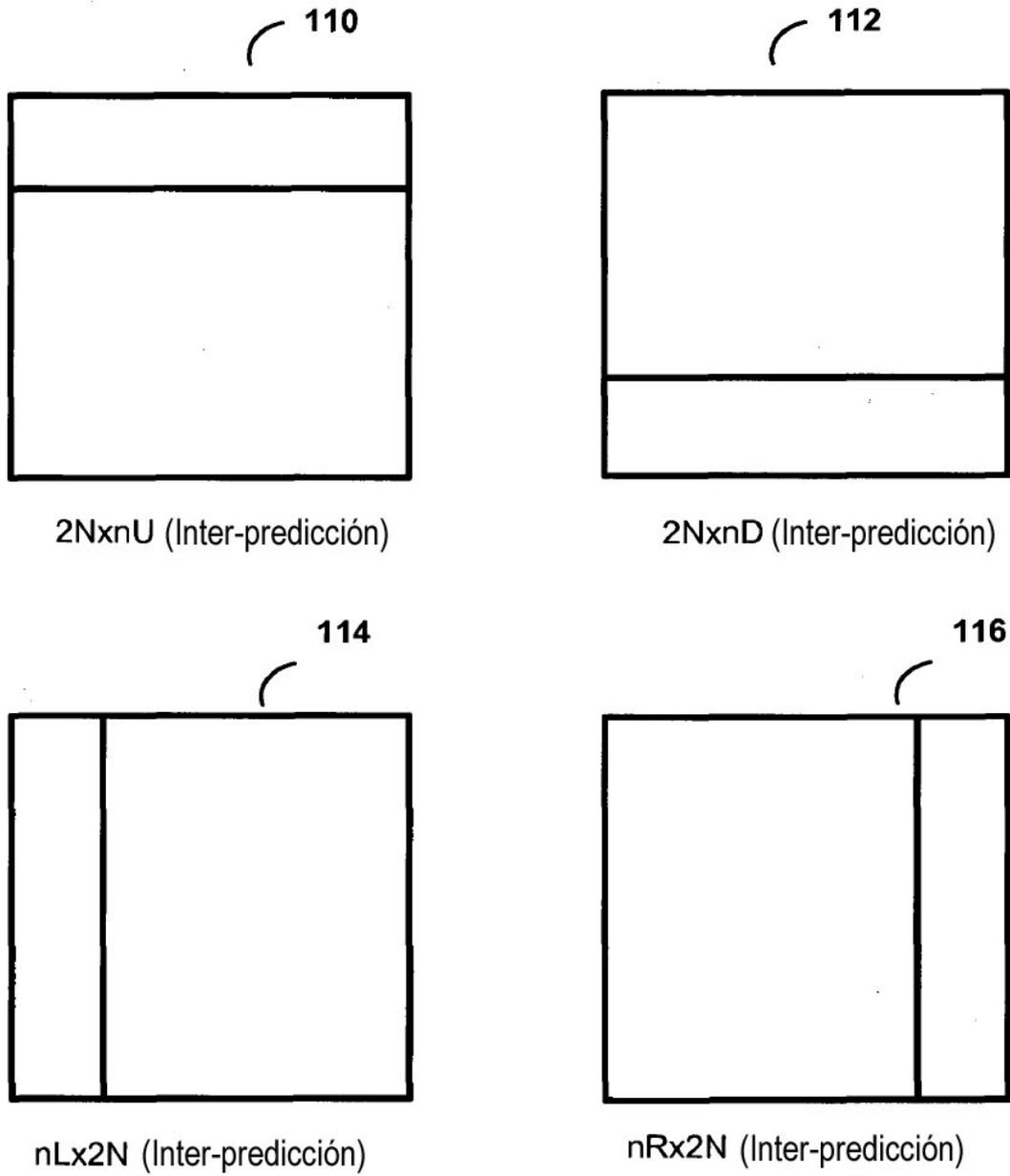


FIG. 5

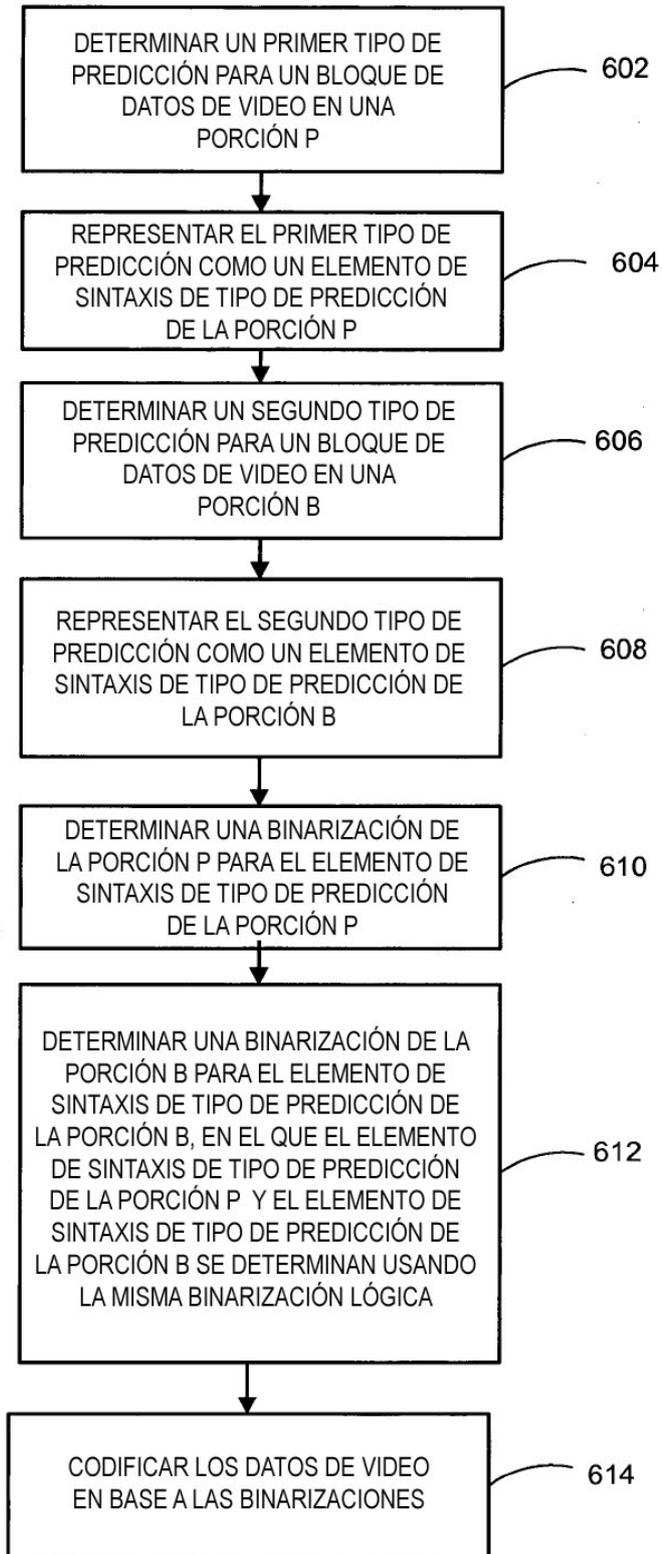


FIG. 6

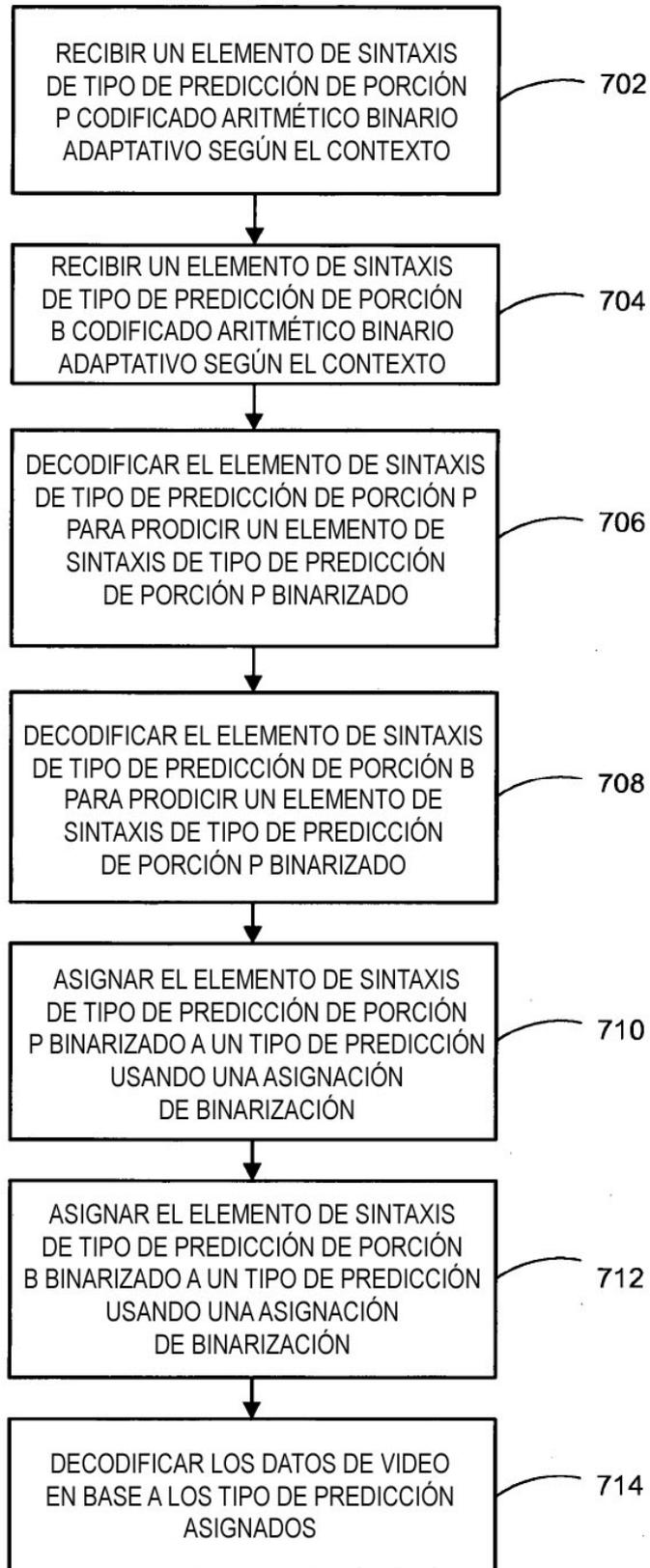


FIG. 7

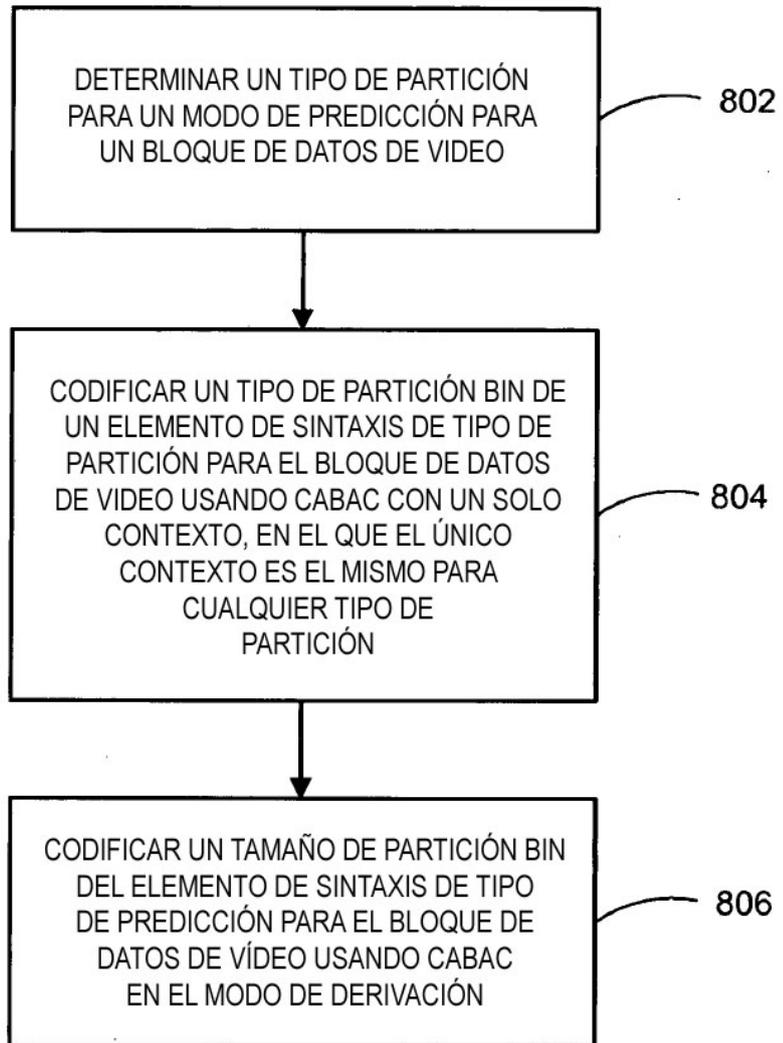


FIG. 8

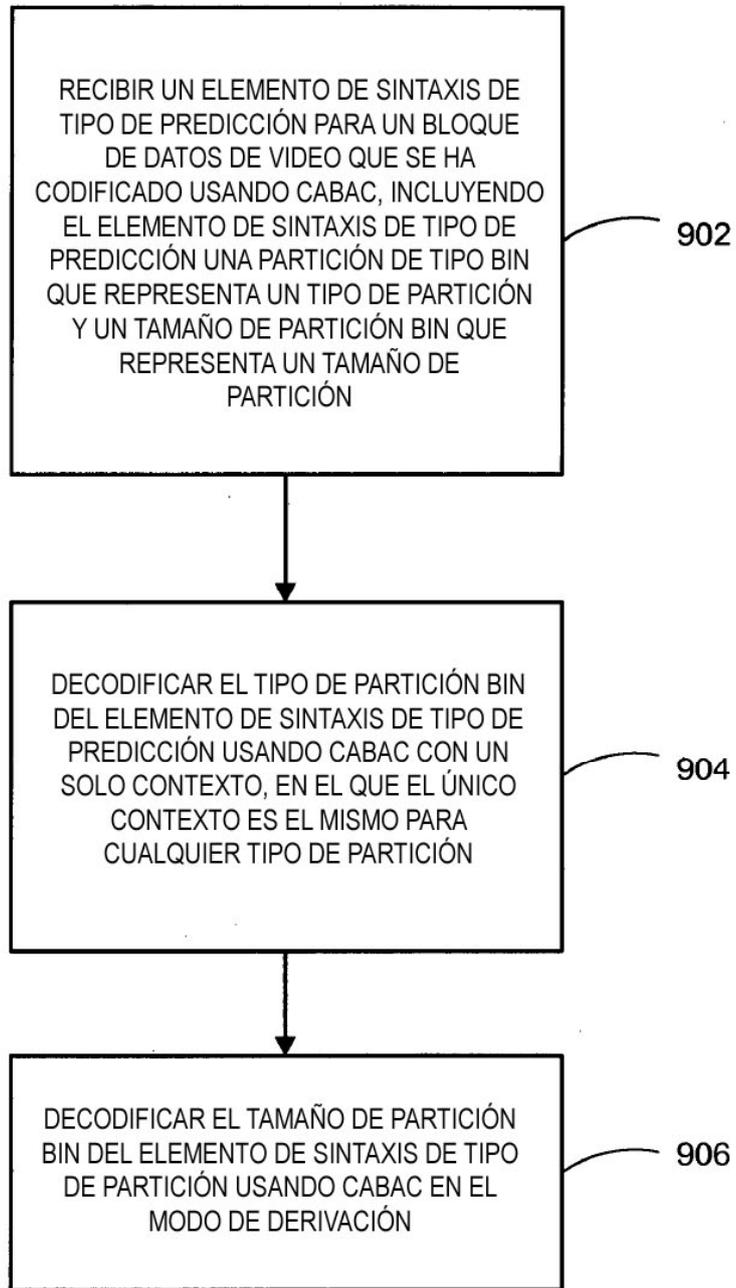


FIG. 9

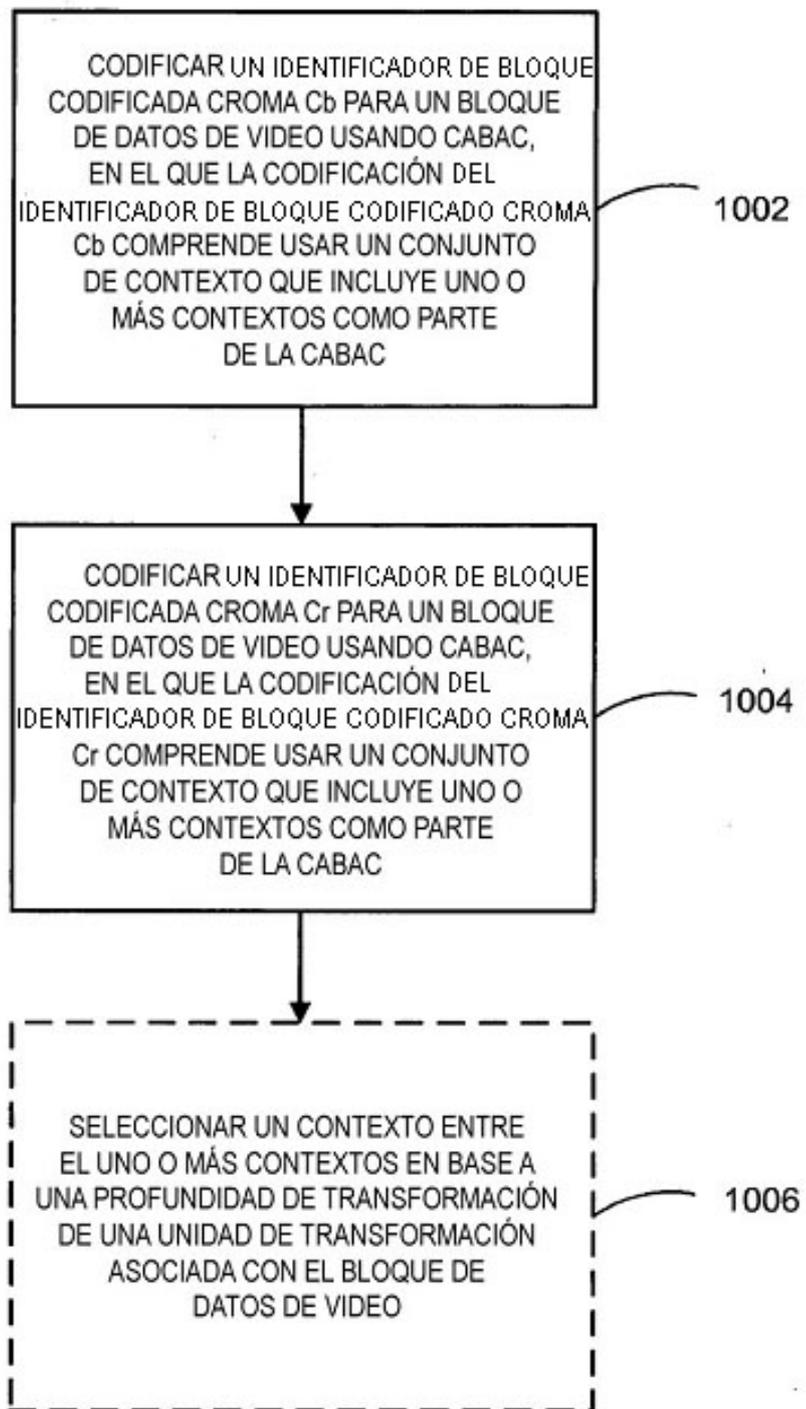


FIG. 10