

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **3 009 058**

51 Int. Cl.:

H04N 19/436 (2014.01)

H04N 19/52 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **10.07.2019 PCT/US2019/041241**

87 Fecha y número de publicación internacional: **16.01.2020 WO20014389**

96 Fecha de presentación y número de la solicitud europea: **10.07.2019 E 19746243 (5)**

97 Fecha y número de publicación de la concesión europea: **08.01.2025 EP 3821604**

54 Título: **Múltiples MVP no adyacentes históricos para el procesamiento de frente de onda de la codificación de video**

30 Prioridad:

10.07.2018 US 201862696281 P

02.08.2018 US 201862713944 P

09.07.2019 US 201916506720

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

25.03.2025

73 Titular/es:

**QUALCOMM INCORPORATED (100.00%)
ATTN: International IP Administration 5775
Morehouse Drive
San Diego, California 92121-1714, US**

72 Inventor/es:

**PHAM VAN, LUONG;
CHIEN, WEI-JUNG;
SEREGIN, VADIM;
KARCZEWICZ, MARTA y
HUANG, HAN**

74 Agente/Representante:

ISERN JARA, Jorge

ES 3 009 058 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

Múltiples MVP no adyacentes históricos para el procesamiento de frente de onda de la codificación de vídeo

5 Campo técnico

Esta descripción se refiere a la codificación de vídeo, que incluye la codificación de vídeo y la decodificación de vídeo.

Antecedentes

10 Las capacidades de vídeo digital pueden incorporarse en una amplia gama de dispositivos, que incluye televisores digitales, sistemas de difusión digital directa, sistemas de difusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, tabletas, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, celular o
 15 teléfonos por radio satélite, los denominados "teléfonos inteligentes", dispositivos de videoconferencia, dispositivos de transmisión de vídeo en directo y similares. Los dispositivos de vídeo digitales implementan técnicas de codificación de vídeo, tales como las descritas en los estándares definidos por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificación de vídeo avanzada (AVC), el estándar de Codificación de vídeo de alta eficiencia (HEVC), ITU-T H.265/Codificación de vídeo de alta eficiencia (HEVC) y extensiones de tales estándares.
 20 Los dispositivos de vídeo pueden transmitir, recibir, codificar, decodificar y/o almacenar información de vídeo digital más eficientemente mediante la implementación de tales técnicas de codificación de vídeo.

25 Las técnicas de codificación de vídeo incluyen la predicción espacial (intraimagen) y/o la predicción temporal (interimagen) para reducir o eliminar la redundancia inherente a las secuencias de vídeo. Para la codificación de vídeo basada en bloques, un segmento de vídeo (por ejemplo, una imagen de vídeo o una porción de una imagen de vídeo) puede dividirse en bloques de vídeo, que también pueden denominarse unidades de árbol de codificación (CTU), unidades de codificación (CU) y/o nodos de codificación. Los bloques de vídeo en un segmento intracodificado (I) de una imagen se codifican mediante el uso de la predicción espacial con respecto a las muestras de referencia de los bloques vecinos de la misma imagen. Los bloques de vídeo en un segmento intercodificado (P o B) de una imagen pueden usar predicción espacial con respecto a las muestras de referencia en bloques vecinos en la misma imagen o predicción temporal con respecto a las muestras de referencia en otras imágenes de referencia. Las imágenes pueden referirse a tramas y las imágenes de referencia pueden referirse como tramas de referencia.

35 La solicitud de patente de Estados Unidos US 2018/0146208 A1 enseña procedimientos y sistemas para la estimación de movimiento restringido por la velocidad en paralelo en un codificador de vídeo donde se determina una lista de al menos dos candidatos a predictores de vectores de movimiento (MVPC), y para al menos dos de estos MVPC se determinan candidatos a vectores de movimiento correspondientes en paralelo mediante el uso de la estimación de movimiento restringida por la velocidad.

40 La solicitud de patente internacional WO 2017/197126 A1 describe un procedimiento para decodificar datos de vídeo que incluye construir una lista de candidatos de vectores de movimiento de candidatos de fusión para el bloque actual de datos de vídeo en base a la información de movimiento de varios bloques vecinos con relación al bloque actual.

Sumario

45 En general, esta descripción describe técnicas para codificar la información de movimiento de bloques de datos de vídeo. Estas técnicas se usan durante el procesamiento paralelo de frente de onda. La información de movimiento puede incluir vectores de movimiento que se predicen a partir de los predictores de vectores de movimiento históricos (HMVP). Un candidato HMVP puede referirse a la información de movimiento de un bloque codificado previamente.
 50 Un codificador de vídeo (codificador o decodificador) puede mantener una tabla con múltiples candidatos HMVP durante un proceso de codificación (codificación o decodificación). El codificador de vídeo puede vaciar la tabla cuando se produce un nuevo segmento. Cuando hay un bloque intercodificado, el codificador de vídeo puede agregar información de movimiento asociada con el bloque intercodificado a la tabla.

55 La presente descripción proporciona un procedimiento de codificación de datos de vídeo de acuerdo con la reivindicación 1, un dispositivo de codificación de datos de vídeo de acuerdo con la reivindicación 11 y un medio de almacenamiento legible por ordenador de acuerdo con la reivindicación 15. Las modalidades específicas son objeto de las reivindicaciones dependientes.

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

Breve descripción de las figuras

65

La Figura 1 es un diagrama de bloques que ilustra un sistema de codificación y decodificación de vídeo ilustrativo que puede realizar las técnicas de esta descripción.

Las Figuras 2A y 2B son diagramas conceptuales que ilustran una estructura de árbol binario de cuádruple (QTBT) de ejemplo, y una unidad de árbol de codificación (CTU) correspondiente.

La Figura 3 es un diagrama de flujo que ilustra un proceso ilustrativo para codificar la información de movimiento mediante el uso de predictores de vectores de movimiento históricos (HMVP).

La Figura 4 es un diagrama conceptual que ilustra un ejemplo de actualización de una tabla HMVP.

La Figura 5 es un diagrama conceptual que ilustra una selección ilustrativa de bloques no adyacentes para la codificación de información de movimiento.

La Figura 6 es un diagrama conceptual que ilustra una selección de ejemplo de bloques no adyacentes en base a un bloque padre.

La Figura 7 es un diagrama conceptual que ilustra un ejemplo de un procesamiento de frente de onda deseado de unidades de árbol de codificación (CTU).

La Figura 8 es un diagrama conceptual que ilustra un ejemplo de información de movimiento usada para HMVP.

La Figura 9 es un diagrama conceptual que ilustra un ejemplo de una imagen dividida en múltiples líneas de unidades de árbol de codificación (CTU).

Las Figuras 10A y 10B son diagramas de bloques que ilustran candidatos de vectores de movimiento vecinos espaciales ilustrativos para los modos de fusión y predicción avanzada de vectores de movimiento (AMVP).

Las Figuras 11A y 11B son diagramas conceptuales que ilustran los candidatos de predicción del vector de movimiento temporal (TMPV).

La Figura 12 es un diagrama de bloques que ilustra un ejemplo de una unidad de árbol de codificación (CTU) y bloques vecinos.

La Figura 13 es un diagrama de bloques que ilustra una CU actual en una CTU actual.

La Figura 14 es un diagrama de bloques que ilustra un codificador de vídeo ilustrativo que puede realizar las técnicas de esta descripción.

La Figura 15 es un diagrama de bloques que ilustra un decodificador de vídeo ilustrativo que puede realizar las técnicas de esta descripción.

La Figura 16 es un diagrama de flujo que ilustra un procedimiento ilustrativo para codificar un bloque actual de datos de vídeo de acuerdo con las técnicas de esta descripción.

La Figura 17 es un diagrama de flujo que ilustra un procedimiento ilustrativo para decodificar un bloque actual de datos de vídeo de acuerdo con las técnicas de esta descripción.

La Figura 18 es un diagrama de flujo que ilustra un procedimiento ilustrativo de codificación (codificación o decodificación) de datos de vídeo de acuerdo con las técnicas de esta descripción.

Descripción detallada

La Figura 1 es un diagrama de bloques que ilustra un sistema de codificación y decodificación de vídeo 100 ilustrativo que puede realizar las técnicas de esta descripción. Las técnicas de esta descripción se dirigen generalmente a codificar (codificar y/o decodificar) datos de vídeo. En general, los datos de vídeo incluyen cualquier dato para procesar un vídeo. Por lo tanto, los datos de vídeo pueden incluir vídeo sin procesar, sin codificar, vídeo codificado, vídeo decodificado (por ejemplo, reconstruido) y metadatos de vídeo, tales como datos de señalización.

Como se muestra en la Figura 1, el sistema 100 incluye un dispositivo fuente 102 que proporciona datos de vídeo codificados para ser decodificados y mostrados por un dispositivo de destino 116, en este ejemplo. En particular, el dispositivo fuente 102 proporciona los datos de vídeo al dispositivo de destino 116 a través de un medio legible por ordenador 110. El dispositivo fuente 102 y el dispositivo de destino 116 pueden comprender cualquiera de una amplia gama de dispositivos, que incluyen computadoras de escritorio, computadoras portátiles (es decir, portátiles), tabletas, decodificadores, teléfonos móviles tales como teléfonos inteligentes, televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de transmisión de vídeo, o similares. En algunos casos, el dispositivo fuente 102 y el dispositivo de destino 116 pueden estar equipados para la comunicación inalámbrica, y por lo tanto pueden denominarse dispositivos de comunicación inalámbrica.

En el ejemplo de la Figura 1, el dispositivo fuente 102 incluye la fuente de vídeo 104, la memoria 106, el codificador de vídeo 200 y la interfaz de salida 108. El dispositivo de destino 116 incluye la interfaz de entrada 122, el decodificador de vídeo 300, la memoria 120 y el dispositivo de visualización 118. De acuerdo con esta descripción, el codificador de vídeo 200 del dispositivo fuente 102 y el decodificador de vídeo 300 del dispositivo de destino 116 pueden configurarse para aplicar las técnicas para codificar la información de movimiento. Por lo tanto, el dispositivo fuente 102 representa un ejemplo de un dispositivo de codificación de vídeo, mientras que el dispositivo de destino 116 representa un ejemplo de un dispositivo de decodificación de vídeo. En otros ejemplos, un dispositivo fuente y un dispositivo de destino pueden incluir otros componentes o disposiciones. Por ejemplo, el dispositivo fuente 102 puede recibir datos de vídeo de una fuente de vídeo externa, tal como una cámara externa. Asimismo, el dispositivo de destino 116 puede interactuar con un dispositivo de visualización externo, en lugar de incluir un dispositivo de visualización integrado.

El sistema 100 como se muestra en la Figura 1 es simplemente un ejemplo. En general, cualquier dispositivo de codificación y/o decodificación de vídeo digital puede realizar técnicas para codificar información de movimiento. El dispositivo fuente 102 y el dispositivo de destino 116 son simplemente ejemplos de tales dispositivos de codificación

en los que el dispositivo fuente 102 genera datos de vídeo codificados para su transmisión al dispositivo de destino 116. Esta descripción se refiere a un dispositivo de "codificación" como un dispositivo que realiza la codificación (codificación y/o decodificación) de datos. Por lo tanto, el codificador de vídeo 200 y el decodificador de vídeo 300 representan ejemplos de dispositivos de codificación, en particular, un codificador de vídeo y un decodificador de vídeo, respectivamente. En algunos ejemplos, el dispositivo fuente 102 y el dispositivo de destino 116 pueden operar de una manera sustancialmente simétrica de manera que cada uno del dispositivo fuente 102 y el dispositivo de destino 116 incluye componentes de codificación y decodificación de vídeo. Por lo tanto, el sistema 100 puede admitir la transmisión de vídeo unidireccional o bidireccional entre el dispositivo fuente 102 y el dispositivo de destino 116, por ejemplo, para transmisión de vídeo, reproducción de vídeo, difusión de vídeo o telefonía de vídeo.

En general, la fuente de vídeo 104 representa una fuente de datos de vídeo (es decir, datos de vídeo sin procesar y sin codificar) y proporciona una serie secuencial de imágenes (también denominadas "tramas") de los datos de vídeo al codificador de vídeo 200, que codifica los datos para las imágenes. La fuente de vídeo 104 del dispositivo fuente 102 puede incluir un dispositivo de captura de vídeo, tal como una cámara de vídeo, un archivo de vídeo que contiene vídeo sin procesar capturado previamente, y/o una interfaz de alimentación de vídeo para recibir vídeo de un proveedor de contenido de vídeo. Como una alternativa adicional, la fuente de vídeo 104 puede generar datos basados en gráficos de ordenador como la fuente de vídeo, o una combinación de vídeo en vivo, vídeo que se archiva y vídeo que se genera por ordenador. En cada caso, el codificador de vídeo 200 codifica los datos de vídeo capturados, precapturados o generados por ordenador. El codificador de vídeo 200 puede reorganizar las imágenes del orden recibido (a veces denominado "orden de visualización") en un orden de codificación para la codificación. El codificador de vídeo 200 puede generar un flujo de bits que incluye datos de vídeo codificados. El dispositivo fuente 102 puede entonces emitir los datos de vídeo codificados a través de la interfaz de salida 108 en un medio legible por ordenador 110 para su recepción y/o recuperación por, por ejemplo, la interfaz de entrada 122 del dispositivo de destino 116.

La memoria 106 del dispositivo fuente 102 y la memoria 120 del dispositivo de destino 116 representan memorias de propósito general. En algunos ejemplos, las memorias 106, 120 pueden almacenar datos de vídeo sin procesar, por ejemplo, vídeo sin procesar de la fuente de vídeo 104 y datos de vídeo sin procesar y decodificados del decodificador de vídeo 300. Adicional o alternativamente, las memorias 106, 120 pueden almacenar instrucciones de software ejecutables por, por ejemplo, el codificador de vídeo 200 y el decodificador de vídeo 300, respectivamente. Aunque se muestran por separado del codificador de vídeo 200 y el decodificador de vídeo 300 en este ejemplo, debe entenderse que el codificador de vídeo 200 y el decodificador de vídeo 300 también pueden incluir memorias internas para propósitos funcionalmente similares o equivalentes. Además, las memorias 106, 120 pueden almacenar datos de vídeo codificados, por ejemplo, salida del codificador de vídeo 200 y entrada al decodificador de vídeo 300. En algunos ejemplos, porciones de las memorias 106, 120 pueden asignarse como uno o más memorias intermedias de vídeo, por ejemplo, para almacenar datos de vídeo sin procesar, decodificados y/o codificados.

El medio legible por ordenador 110 puede representar cualquier tipo de medio o dispositivo capaz de transportar los datos de vídeo codificados desde el dispositivo fuente 102 al dispositivo de destino 116. En un ejemplo, el medio legible por ordenador 110 representa un medio de comunicación para permitir que el dispositivo fuente 102 transmita datos de vídeo codificados directamente al dispositivo de destino 116 en tiempo real, por ejemplo, a través de una red de radiofrecuencia o una red basada en ordenador. La interfaz de salida 108 puede modular una señal de transmisión que incluye los datos de vídeo codificados, y la interfaz de entrada 122 puede demodular la señal de transmisión recibida, de acuerdo con un estándar de comunicación, tal como un protocolo de comunicación inalámbrica. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrica o alámbrica, tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión física. El medio de comunicación puede formar parte de una red en base a paquetes, tal como una red de área local, una red de área amplia o una red global tal como Internet. El medio de comunicación puede incluir enrutadores, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo fuente 102 al dispositivo de destino 116.

En algunos ejemplos, el dispositivo fuente 102 puede emitir datos codificados desde la interfaz de salida 108 al dispositivo de almacenamiento 112. De manera similar, el dispositivo de destino 116 puede acceder a los datos codificados desde el dispositivo de almacenamiento 112 a través de la interfaz de entrada 122. El dispositivo de almacenamiento 112 puede incluir cualquiera de una variedad de medios de almacenamiento de datos distribuidos o de acceso local, tal como un disco duro, discos Blu-ray, DVD, CD-ROM, memoria flash, memoria volátil o no volátil, o cualquier otro medio de almacenamiento digital adecuado para almacenar datos de vídeo codificados.

En algunos ejemplos, el dispositivo fuente 102 puede enviar datos de vídeo codificados al servidor de archivos 114 u otro dispositivo de almacenamiento intermedio que puede almacenar el vídeo codificado generado por el dispositivo fuente 102. El dispositivo de destino 116 puede acceder a los datos de vídeo almacenados desde el servidor de archivos 114 a través de transmisión continua o descarga. El servidor de archivos 114 puede ser cualquier tipo de dispositivo servidor capaz de almacenar datos de vídeo codificados y transmitir esos datos de vídeo codificados al dispositivo de destino 116. El servidor de archivos 114 puede representar un servidor web (por ejemplo, para un sitio web), un servidor de Protocolo de transferencia de archivos (FTP), un dispositivo de red de entrega de contenido o un dispositivo de almacenamiento conectado a la red (NAS). El dispositivo de destino 116 puede acceder a datos de vídeo codificados desde el servidor de archivos 114 a través de cualquier conexión de datos estándar, que incluye una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión Wi-Fi), una conexión por

cable (por ejemplo, línea de abonado digital (DSL), módem de cable, etc.), o una combinación de ambos que sea adecuada para acceder a datos de vídeo codificados almacenados en el servidor de archivos 114. El servidor de archivos 114 y la interfaz de entrada 122 pueden configurarse para operar de acuerdo con un protocolo de transmisión en tiempo real, un protocolo de transmisión de descarga, o una de sus combinaciones.

La interfaz de salida 108 y la interfaz de entrada 122 pueden representar transmisores/receptores inalámbricos, módems, componentes de red cableados (por ejemplo, tarjetas Ethernet), componentes de comunicación inalámbrica que funcionan de acuerdo con cualquiera de una variedad de estándares IEEE 802.11, u otros componentes físicos. En ejemplos donde la interfaz de salida 108 y la interfaz de entrada 122 comprenden componentes inalámbricos, la interfaz de salida 108 y la interfaz de entrada 122 pueden configurarse para transferir datos, tales como datos de vídeo codificados, de acuerdo con un estándar de comunicación celular, tal como 4G, 4G-LTE (evolución a largo plazo), LTE Advanced, 5G, o similares. En algunos ejemplos donde la interfaz de salida 108 comprende un transmisor inalámbrico, la interfaz de salida 108 y la interfaz de entrada 122 pueden configurarse para transferir datos, tales como datos de vídeo codificados, de acuerdo con otros estándares inalámbricos, tales como una especificación IEEE 802.11, una especificación IEEE 802.15 (por ejemplo, ZigBee™), un Bluetooth™ estándar, o similares. En algunos ejemplos, el dispositivo fuente 102 y/o el dispositivo de destino 116 pueden incluir dispositivos respectivos de sistema en un chip (SoC). Por ejemplo, el dispositivo fuente 102 puede incluir un dispositivo SoC para realizar la funcionalidad atribuida al codificador de vídeo 200 y/o la interfaz de salida 108, y el dispositivo de destino 116 puede incluir un dispositivo SoC para realizar la funcionalidad atribuida al decodificador de vídeo 300 y/o la interfaz de entrada 122.

Las técnicas de esta descripción pueden aplicarse a la codificación de vídeo en apoyo de cualquiera de una variedad de aplicaciones multimedia, tales como difusiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo en tiempo real por Internet, tales como transmisión adaptable y dinámica sobre HTTP (DASH), vídeo digital que se codifica en un medio de almacenamiento de datos, decodificación de vídeo digital almacenado en un medio de almacenamiento de datos, u otras aplicaciones.

La interfaz de entrada 122 del dispositivo de destino 116 recibe un flujo de bits de vídeo codificado desde el medio legible por ordenador 110 (por ejemplo, el dispositivo de almacenamiento 112, el servidor de archivos 114, o similares). El flujo de bits de vídeo codificado puede incluir información de señalización definida por el codificador de vídeo 200, que también usa el decodificador de vídeo 300, tal como elementos de sintaxis que tienen valores que describen las características y/o el procesamiento de bloques de vídeo u otras unidades codificadas (por ejemplo, segmentos, imágenes, grupos de imágenes, secuencias, o similares). El dispositivo de visualización 118 muestra imágenes decodificadas de los datos de vídeo decodificados a un usuario. El dispositivo de visualización 118 puede representar cualquiera de una variedad de dispositivos de visualización tales como un tubo de rayos catódicos (CRT), una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodo emisor de luz orgánico (OLED) u otro tipo de dispositivo de visualización.

Aunque no se muestra en la Figura 1, en algunos ejemplos, el codificador de vídeo 200 y el decodificador de vídeo 300 pueden integrarse cada uno con un codificador de audio y/o un decodificador de audio, y pueden incluir unidades MUX-DEMUX apropiadas, u otro hardware y/o software, para manejar flujos multiplexados que incluyen tanto audio como vídeo en un flujo de datos común. Si corresponde, las unidades MUX-DEMUX se pueden ajustar al protocolo multiplexor ITU H.223, o a otros protocolos tales como el protocolo de datagramas de usuario (UDP).

El codificador de vídeo 200 y el decodificador de vídeo 300 pueden implementarse cada uno como cualquiera de una variedad de circuitos codificadores y/o decodificadores adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados de aplicación específica (ASIC), matrices de puertas programables en campo (FPGA), lógica discreta, software, hardware, microprograma o cualquiera de sus combinaciones. Cuando las técnicas se implementan de manera parcial en el software, un dispositivo puede almacenar instrucciones para el software en un medio legible por ordenador no transitorio adecuado y ejecutar las instrucciones en el hardware mediante el uso de uno o más procesadores para realizar las técnicas de esta divulgación. Cada uno de los codificadores de vídeo 200 y el decodificador de vídeo 300 puede incluirse en uno o más codificadores o decodificadores, cualquiera de los cuales puede integrarse como parte de un codificador/decodificador combinado (CODEC) en un dispositivo respectivo. Un dispositivo que incluye un codificador de vídeo 200 y/o un decodificador de vídeo 300 puede comprender un circuito integrado, un microprocesador, y/o un dispositivo de comunicación inalámbrica, tal como un teléfono celular.

El codificador de vídeo 200 y el decodificador de vídeo 300 pueden funcionar de acuerdo con un estándar de codificación de vídeo, tal como ITU-T H.265, también denominado Codificación de vídeo de alta eficiencia (HEVC) o extensiones a la misma, tales como las extensiones de codificación de vídeo multivista y/o escalable. Alternativamente, el codificador de vídeo 200 y el decodificador de vídeo 300 pueden operar de acuerdo con otros estándares industriales o patentados, tales como el modelo de prueba de exploración conjunta (JEM) o ITU-T H.266, también denominado codificación de vídeo versátil (VVC). Un borrador reciente del estándar VVC se describe en Bross, y otros. "Versatile Video Coding (Draft 5)," Equipo de Expertos en Vídeo Conjunto (JVET) de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 14a reunión: Ginebra, CH, 19-27 de marzo de 2019, JVET-N1001-v3 (en adelante "Borrador 5 de VVC"). Las técnicas de esta divulgación, sin embargo, no se limitan a ningún estándar de codificación particular.

En general, el codificador de vídeo 200 y el decodificador de vídeo 300 pueden realizar la codificación basada en bloques de imágenes. El término "bloque" generalmente se refiere a una estructura que incluye datos a procesar (por ejemplo, codificados, decodificados o usados de otro modo en el proceso de codificación y/o decodificación). Por ejemplo, un bloque puede incluir una matriz bidimensional de muestras de datos de luminancia y/o crominancia. En general, el codificador de vídeo 200 y el decodificador de vídeo 300 pueden codificar datos de vídeo representados en un formato YUV (por ejemplo, Y, Cb, Cr). Es decir, en lugar de codificar datos rojo, verde y azul (RGB) para muestras de una imagen, el codificador de vídeo 200 y el decodificador de vídeo 300 pueden codificar componentes de luminancia y crominancia, donde los componentes de crominancia pueden incluir componentes de crominancia de matiz rojo y matiz azul. En algunos ejemplos, el codificador de vídeo 200 convierte los datos recibidos con formato RGB a una representación YUV antes de la codificación, y el decodificador de vídeo 300 convierte la representación YUV al formato RGB. Alternativamente, las unidades de pre y post-procesamiento (no mostradas) pueden realizar estas conversiones.

Esta descripción puede referirse generalmente a la codificación (por ejemplo, codificación y decodificación) de imágenes para incluir el proceso de codificación o decodificación de datos de la imagen. De manera similar, esta descripción puede referirse a la codificación de bloques de una imagen para incluir el proceso de codificación o decodificación de datos para los bloques, por ejemplo, predicción y/o codificación residual. Un flujo de bits de vídeo codificado generalmente incluye una serie de valores para elementos de sintaxis representativos de decisiones de codificación (por ejemplo, modos de codificación) y partición de imágenes en bloques. Por lo tanto, las referencias a la codificación de una imagen o un bloque deben entenderse generalmente como la codificación de valores para elementos de sintaxis que forman la imagen o el bloque.

HEVC define varios bloques, que incluyen unidades de codificación (CU), unidades de predicción (PU) y unidades de transformación (TU). De acuerdo con HEVC, un codificador de vídeo (tal como el codificador de vídeo 200) divide una unidad de árbol de codificación (CTU) en CU de acuerdo con una estructura de árbol cuaternario. Es decir, el codificador de vídeo divide las CTU y las CU en cuatro cuadrados iguales y no solapados, y cada nodo del árbol cuaternario tiene cero o cuatro nodos hijos. Los nodos sin nodos hijos pueden denominarse "nodos hoja", y las CU de tales nodos hoja pueden incluir una o más PU y/o una o más TU. El codificador de vídeo puede dividir además las PU y TU. Por ejemplo, en HEVC, un árbol cuaternario residual (RQT) representa la partición de TU. En HEVC, las PU representan datos de interpredicción, mientras que las TU representan datos residuales. Las CU que se intrapredicen incluyen información de intrapredicción, tal como una indicación de intramodo.

Como otro ejemplo, el codificador de vídeo 200 y el decodificador de vídeo 300 pueden configurarse para operar de acuerdo con JEM o VVC. De acuerdo con JEM o VVC, un codificador de vídeo (tal como el codificador de vídeo 200) divide una imagen en una pluralidad de unidades de árbol de codificación (CTU). El codificador de vídeo 200 puede dividir una CTU de acuerdo con una estructura de árbol, tal como una estructura de árbol cuaternario-árbol binario (QTBT) o una estructura de árbol de múltiples tipos (MTT). La estructura QTBT elimina los conceptos de múltiples tipos de partición, tales como la separación entre CU, PU y TU de HEVC. Una estructura QTBT puede incluir dos niveles: un primer nivel dividido de acuerdo con la partición de árbol cuaternario y un segundo nivel dividido de acuerdo con la partición de árbol binario. Un nodo raíz de la estructura QTBT corresponde a una CTU. Los nodos hoja de los árboles binarios corresponden a unidades de codificación (CU).

En una estructura de partición MTT, los bloques pueden dividirse mediante el uso de una partición de árbol cuaternario (QT), una partición de árbol binario (BT) y uno o más tipos de particiones de árbol triple (TT). Una partición de árbol triple es una partición donde un bloque se divide en tres subbloques. En algunos ejemplos, una partición de árbol triple divide un bloque en tres subbloques sin dividir el bloque original a través del centro. Los tipos de partición en MTT (por ejemplo, QT, BT y TT) pueden ser simétricos o asimétricos.

En algunos ejemplos, el codificador de vídeo 200 y el decodificador de vídeo 300 pueden usar una única estructura QTBT o MTT para representar cada uno de los componentes de luminancia y crominancia, mientras que en otros ejemplos, el codificador de vídeo 200 y el decodificador de vídeo 300 pueden usar dos o más estructuras QTBT o MTT, tales como una estructura QTBT o MTT para el componente de luminancia y otra estructura QTBT o MTT para ambos componentes de crominancia (o dos estructuras QTBT o MTT para componentes de crominancia respectivos).

El codificador de vídeo 200 y el decodificador de vídeo 300 pueden configurarse para usar la partición de árbol cuaternario por HEVC, la partición QTBT de acuerdo con JEM u otras estructuras de partición. Para fines de explicación, la descripción de las técnicas de esta descripción se presenta con respecto a la partición QTBT. Sin embargo, debe entenderse que las técnicas de esta descripción también pueden aplicarse a codificadores de vídeo configurados para usar particiones de árbol cuaternario, o también otros tipos de particiones.

Esta descripción puede usar "NxN" y "N por N" indistintamente para referirse a las dimensiones de muestra de un bloque (tal como una CU u otro bloque de vídeo) en términos de dimensiones verticales y horizontales, por ejemplo, 16x16 muestras o 16 por 16 muestras. En general, una CU 16x16 tendrá 16 muestras en una dirección vertical (y = 16) y 16 muestras en una dirección horizontal (x = 16). Igualmente, una CU NxN generalmente tiene N muestras en una dirección vertical y N muestras en una dirección horizontal, donde N representa un valor entero no negativo. Las muestras en una CU pueden disponerse en hileras y columnas. Además, las CU no necesitan tener necesariamente

el mismo número de muestras en la dirección horizontal que en la dirección vertical. Por ejemplo, las CU pueden comprender muestras $N \times M$, donde M no es necesariamente igual a N .

5 El codificador de vídeo 200 codifica los datos de vídeo para las CU que representan la información de predicción y/o residual, y otra información. La información de predicción indica cómo se va a predecir la CU para formar un bloque de predicción para la CU. La información residual generalmente representa diferencias muestra por muestra entre muestras de la CU antes de la codificación y el bloque de predicción.

10 Para predecir una CU, el codificador de vídeo 200 puede formar generalmente un bloque de predicción para la CU mediante la interpredicción o la intrapredicción. La interpredicción generalmente se refiere a predecir la CU a partir de los datos de una imagen previamente codificada, mientras que la intrapredicción generalmente se refiere a predecir la CU a partir de los datos previamente codificados de la misma imagen. Para realizar la interpredicción, el codificador de vídeo 200 puede generar el bloque de predicción mediante el uso de uno o más vectores de movimiento. El
15 codificador de vídeo 200 puede realizar generalmente una búsqueda de movimiento para identificar un bloque de referencia que coincida estrechamente con la CU, por ejemplo, en términos de diferencias entre la CU y el bloque de referencia. El codificador de vídeo 200 puede calcular una métrica de diferencia mediante el uso de una suma de diferencia absoluta (SAD), suma de diferencias al cuadrado (SSD), diferencia absoluta media (MAD), diferencias al cuadrado media (MSD) u otros cálculos de diferencia similares para determinar si un bloque de referencia coincide estrechamente con la CU actual. En algunos ejemplos, el codificador de vídeo 200 puede predecir la CU actual
20 mediante el uso de predicción unidireccional o predicción bidireccional.

JEM también proporciona un modo de compensación de movimiento afín, que puede considerarse un modo de interpredicción. En el modo de compensación de movimiento afín, el codificador de vídeo 200 puede determinar dos
25 o más vectores de movimiento que representan el movimiento no traslacional, tal como acercamiento o alejamiento, rotación, movimiento en perspectiva u otros tipos de movimiento irregulares.

Para realizar la intrapredicción, el codificador de vídeo 200 puede seleccionar un modo de intrapredicción para generar el bloque de predicción. JEM proporciona sesenta y siete modos de intrapredicción, que incluyen varios modos
30 direccionales, así como también modo plano y modo DC. En general, el codificador de vídeo 200 selecciona un modo de intrapredicción que describe las muestras vecinas de un bloque actual (por ejemplo, un bloque de una CU) a partir del cual predecir muestras del bloque actual. Tales muestras generalmente pueden estar por encima, por encima y a la izquierda, o a la izquierda del bloque actual en la misma imagen que el bloque actual, al asumir que el codificador de vídeo 200 codifica las CTU y CU en el orden de barrido de trama (de izquierda a derecha, de arriba a abajo).

35 El codificador de vídeo 200 codifica los datos que representan el modo de predicción para un bloque actual. Por ejemplo, para los modos de interpredicción, el codificador de vídeo 200 puede codificar datos que representan cuál de los diversos modos de interpredicción disponibles se usa, así como también información de movimiento para el modo correspondiente. Para la interpredicción unidireccional o bidireccional, por ejemplo, el codificador de vídeo 200 puede codificar vectores de movimiento mediante el uso de la predicción avanzada de vectores de movimiento (AMVP) o el
40 modo de fusión. El codificador de vídeo 200 puede usar modos similares para codificar vectores de movimiento para el modo de compensación de movimiento afín.

Después de la predicción, tal como la intrapredicción o la interpredicción de un bloque, el codificador de vídeo 200
45 puede calcular los datos residuales para el bloque. Los datos residuales, tales como un bloque residual, representan diferencias muestra por muestra entre el bloque y un bloque de predicción para el bloque, formado mediante el uso del modo de predicción correspondiente. El codificador de vídeo 200 puede aplicar una o más transformaciones al bloque residual, para producir datos transformados en un dominio de transformación en lugar del dominio de muestra. Por ejemplo, el codificador de vídeo 200 puede aplicar una transformada discreta del coseno (DCT), una transformada entera, una transformada de ondículas o una transformada conceptualmente similar a los datos de vídeo residuales.
50 Adicionalmente, el codificador de vídeo 200 puede aplicar una transformada secundaria después de la primera transformada, tal como una transformada secundaria no separable dependiente del modo (MDNSST), una transformada dependiente de la señal, una transformada de Karhunen-Loeve (KLT), o similar. El codificador de vídeo 200 produce coeficientes de transformación después de la aplicación de una o más transformaciones.

55 Como se indicó anteriormente, después de cualquier transformación para producir coeficientes de transformación, el codificador de vídeo 200 puede realizar la cuantificación de los coeficientes de transformación. La cuantificación de manera general se refiere a un procedimiento en el que los coeficientes de transformación se cuantifican para de manera posible reducir la cantidad de datos utilizados para representar los coeficientes, proporcionando compresión adicional. Al realizar el proceso de cuantificación, el codificador de vídeo 200 puede reducir la profundidad de bits
60 asociada con algunos o todos los coeficientes. Por ejemplo, el codificador de vídeo 200 puede redondear un n -valor de bits hasta un m -valor de bits durante la cuantificación, donde n es mayor que m . En algunos ejemplos, para realizar la cuantificación, el codificador de vídeo 200 puede realizar un desplazamiento a la derecha bit a bit del valor a cuantificar.

65 Después de la cuantificación, el codificador de vídeo 200 puede escanear los coeficientes de transformación, lo que produce un vector unidimensional de la matriz bidimensional que incluye los coeficientes de transformación

cuantificados. El escaneo puede diseñarse para colocar coeficientes de energía más alta (y por lo tanto de frecuencia más baja) en la parte delantera del vector y para colocar coeficientes de transformación de energía más baja (y por lo tanto de frecuencia más alta) en la parte trasera del vector. En algunos ejemplos, el codificador de vídeo 200 puede utilizar un orden de barrido predefinido para barrer los coeficientes de transformación cuantificados para producir un vector serializado, y después codificar por entropía los coeficientes de transformación cuantificados del vector. En otros ejemplos, el codificador de vídeo 200 puede realizar un escaneo adaptativo. Después de escanear los coeficientes de transformación cuantificados para formar el vector unidimensional, el codificador de vídeo 200 puede codificar por entropía el vector unidimensional, por ejemplo, de acuerdo con la codificación aritmética binaria adaptativa al contexto (CABAC). El codificador de vídeo 200 también puede codificar por entropía valores para elementos de sintaxis que describen los metadatos asociados con los datos de vídeo codificados para su uso por el decodificador de vídeo 300 en la decodificación de los datos de vídeo.

Para realizar la CABAC, el codificador de vídeo 200 puede asignar un contexto dentro de un modelo de contexto a un símbolo a transmitir. El contexto puede referirse, por ejemplo, a si los valores vecinos del símbolo tienen un valor cero o no. La determinación de la probabilidad puede ser en base a un contexto asignado al símbolo.

El codificador de vídeo 200 puede generar además datos de sintaxis, tales como datos de sintaxis basados en bloques, datos de sintaxis basados en imágenes y datos de sintaxis basados en secuencias, al decodificador de vídeo 300, por ejemplo, en un encabezado de imagen, un encabezado de bloque, un encabezado de segmento, u otros datos de sintaxis, tales como un conjunto de parámetros de secuencia (SPS), un conjunto de parámetros de imagen (PPS) o un conjunto de parámetros de vídeo (VPS). El decodificador de vídeo 300 también puede decodificar tales datos de sintaxis para determinar cómo decodificar los datos de vídeo correspondientes.

De esta manera, el codificador de vídeo 200 puede generar un flujo de bits que incluye datos de vídeo codificados, por ejemplo, elementos de sintaxis que describen la partición de una imagen en bloques (por ejemplo, CU) e información de predicción y/o residual para los bloques. En última instancia, el decodificador de vídeo 300 puede recibir el flujo de bits y decodificar los datos de vídeo codificados.

En general, el decodificador de vídeo 300 realiza un proceso recíproco al realizado por el codificador de vídeo 200 para decodificar los datos de vídeo codificados del flujo de bits. Por ejemplo, el decodificador de vídeo 300 puede decodificar valores para elementos de sintaxis del flujo de bits mediante el uso de CABAC de una manera sustancialmente similar, aunque recíproca, al proceso de codificación CABAC del codificador de vídeo 200. Los elementos de sintaxis pueden definir la información de partición de una imagen en CTU, y la partición de cada CTU de acuerdo con una estructura de partición correspondiente, tal como una estructura QTBT, para definir las CU de la CTU. Los elementos de sintaxis pueden definir además la predicción y la información residual para bloques (por ejemplo, CU) de datos de vídeo.

La información residual puede representarse, por ejemplo, mediante coeficientes de transformación cuantificados. El decodificador de vídeo 300 puede cuantificar inversamente y transformar inversamente los coeficientes de transformación cuantificados de un bloque para reproducir un bloque residual para el bloque. El decodificador de vídeo 300 usa un modo de predicción señalado (intra o interpredicción) e información de predicción relacionada (por ejemplo, información de movimiento para la interpredicción) para formar un bloque de predicción para el bloque. El decodificador de vídeo 300 puede combinar el bloque de predicción y el bloque residual (sobre una base de muestra por muestra) para reproducir el bloque original. El decodificador de vídeo 300 puede realizar un procesamiento adicional, tal como realizar un proceso de desbloqueo para reducir los artefactos visuales a lo largo de los límites del bloque.

Esta descripción puede referirse generalmente a "señalar" cierta información, tal como elementos de sintaxis. El término "señalización" puede referirse generalmente a la comunicación de valores para elementos de sintaxis y/u otros datos usados para decodificar datos de vídeo codificados. Es decir, el codificador de vídeo 200 puede señalar valores para elementos de sintaxis en el flujo de bits. En general, la señalización se refiere a generar un valor en el flujo de bits. Como se indicó anteriormente, el dispositivo fuente 102 puede transportar el flujo de bits al dispositivo de destino 116 sustancialmente en tiempo real, o no en tiempo real, tal como podría ocurrir cuando se almacenan elementos de sintaxis en el dispositivo de almacenamiento 112 para su posterior recuperación por el dispositivo de destino 116.

De acuerdo con las técnicas de esta descripción, el codificador de vídeo 200 y el decodificador de vídeo 300 se configuran para realizar el procesamiento paralelo de frente de onda cuando se codifica una imagen de datos de vídeo. En general, el procesamiento paralelo de frentes de onda puede implicar la codificación de líneas individuales de unidades de árbol de codificación (CTU) mediante el uso de subprocesos de procesamiento separados. Por ejemplo, un primer hilo ejecutado por el codificador de vídeo 200 o el decodificador de vídeo 300 puede procesar una primera línea de CTU, un segundo hilo puede procesar una segunda línea de CTU, y así sucesivamente. La codificación de una CTU incluye, entre otras cosas, la codificación de la información de movimiento para las unidades de codificación (CU) de la CTU predicha por movimiento, que puede referirse a la información de movimiento dentro de la misma CTU o una CTU codificada previamente (por ejemplo, una CTU izquierda y/o superior vecina). Tal información de movimiento puede almacenarse en una memoria intermedia del predictor de vectores de movimiento (MVP). De acuerdo con las técnicas de esta descripción, el codificador de vídeo 200 y el decodificador de vídeo 300 se configuran

para restablecer una memoria intermedia de MVP para una línea CTU actual antes de codificar los datos de vídeo de la línea CTU actual. La memoria intermedia de MVP puede ser una memoria intermedia de MVP individual para la línea CTU actual, o puede usarse una memoria intermedia de MVP común para múltiples líneas de CTU.

5 En algunos ejemplos, cuando se almacena información de movimiento en una memoria intermedia de MVP, el codificador de vídeo 200 y el decodificador de vídeo 300 pueden almacenar solo información de movimiento única dentro de la memoria intermedia de MVP. Por ejemplo, el codificador de vídeo 200 y el decodificador de vídeo 300 pueden codificar una CU actual mediante el uso de un vector de movimiento actual, determinar si el vector de movimiento se almacena actualmente en una memoria intermedia de MVP para la CU actual, y si es así, evitar almacenar el vector de movimiento en la memoria intermedia de MVP, y si no, almacenar el vector de movimiento en la memoria intermedia de MVP.

15 En algunos ejemplos, el codificador de vídeo 200 y el decodificador de vídeo 300 pueden usar una regla de primero en entrar primero en salir (FIFO) para eliminar vectores de movimiento de la memoria intermedia de MVP cuando la memoria intermedia de MVP se llena. Es decir, para añadir un nuevo vector de movimiento a la memoria intermedia de MVP, el codificador de vídeo 200 y el decodificador de vídeo 300 pueden eliminar un vector de movimiento insertado más temprano de la memoria intermedia de MVP e insertar el nuevo vector de movimiento en la memoria intermedia de MVP. De esta manera, el amortiguador MVP puede tener un comportamiento similar a una cola.

20 En algunos ejemplos, el codificador de vídeo 200 y el decodificador de vídeo 300 pueden mantener memorias intermedias de MVP separados para cada uno de una variedad de tipos de modelos de movimiento diferentes. Por ejemplo, el codificador de vídeo 200 y el decodificador de vídeo 300 pueden mantener una memoria intermedia de MVP afín para el modelo de movimiento afín, una memoria intermedia de MVP de copia intrabloque para la información de movimiento del modo de copia intrabloque, una memoria intermedia de MVP de compensación de iluminación para la información de movimiento de compensación de iluminación local, una memoria intermedia de MVP de subbloque para MVP de subbloque, y/o una memoria intermedia de MVP temporal para la predicción de movimiento temporal.

30 En algunos ejemplos, el codificador de vídeo 200 y el decodificador de vídeo 300 pueden generar un MVP sintético a partir de dos o más MVP en uno o más memorias intermedias de MVP e insertar el MVP sintético en uno de las memorias intermedias de MVP. Los dos o más MVP pueden ajustarse al mismo o diferentes modelos de movimiento (es decir, tener diferentes tipos de información de movimiento).

35 Las Figuras 2A y 2B son diagramas conceptuales que ilustran una estructura de árbol binario de cuádruple (QTBT) 130 de ejemplo, y una unidad de árbol de codificación (CTU) 132 correspondiente. Las líneas continuas representan la división del árbol cuaternario, y las líneas punteadas indican la división del árbol binario. En cada nodo dividido (es decir, no hoja) del árbol binario, se señala un indicador para indicar qué tipo de división (es decir, horizontal o vertical) se usa, donde 0 indica la división horizontal y 1 indica la división vertical en este ejemplo. Para la división del árbol cuaternario, no es necesario indicar el tipo de división, ya que los nodos del árbol cuaternario dividen un bloque horizontal y verticalmente en 4 subbloques de igual tamaño. En consecuencia, el codificador de vídeo 200 puede codificar, y el decodificador de vídeo 300 puede decodificar, elementos de sintaxis (tales como información de división) para un nivel de árbol de región de la estructura QTBT 130 (es decir, las líneas continuas) y elementos de sintaxis (tales como información de división) para un nivel de árbol de predicción de la estructura QTBT 130 (es decir, las líneas discontinuas). El codificador de vídeo 200 puede codificar, y el decodificador de vídeo 300 puede decodificar, datos de vídeo, tales como datos de predicción y transformación, para las CU representadas por los nodos hoja terminales de la estructura QTBT 130.

50 En general, la CTU 132 de la Figura 2B puede asociarse con parámetros que definen los tamaños de los bloques correspondientes a los nodos de la estructura QTBT 130 en el primer y segundo niveles (por ejemplo, nivel de árbol de región y nivel de árbol de predicción). Estos parámetros pueden incluir un tamaño de CTU (que representa un tamaño de CTU 132 en muestras), un tamaño de árbol cuaternario mínimo (MinQTSIZE, que representa un tamaño de nodo hoja de árbol cuaternario mínimo permitido), un tamaño de árbol binario máximo (MaxBTSIZE, que representa un tamaño de nodo raíz de árbol binario máximo permitido), una profundidad de árbol binario máxima (MaxBTDEPTH, que representa una profundidad de árbol binario máxima permitida), y un tamaño de árbol binario mínimo (MinBTSIZE, que representa el tamaño de nodo hoja de árbol binario mínimo permitido).

55 El nodo raíz de una estructura QTBT correspondiente a una CTU puede tener cuatro nodos hijos en el primer nivel de la estructura QTBT, cada uno de los cuales puede dividirse de acuerdo con la partición de árbol cuaternario. Es decir, los nodos del primer nivel son nodos hoja (que no tienen nodos hijos) o tienen cuatro nodos hijos. El ejemplo de la estructura QTBT 130 representa tales nodos como que incluyen el nodo padre y los nodos hijos que tienen líneas sólidas para las ramas. Si los nodos del primer nivel no son más grandes que el tamaño máximo permitido del nodo raíz del árbol binario (MaxBTSIZE), entonces los nodos pueden dividirse adicionalmente en árboles binarios respectivos. La división del árbol binario de un nodo puede iterarse hasta que los nodos resultantes de la división alcancen el tamaño mínimo de nodo hoja del árbol binario permitido (MinBTSIZE) o la profundidad máxima del árbol binario permitida (MaxBTDEPTH). El ejemplo de la estructura QTBT 130 representa tales nodos como que tienen líneas discontinuas para las ramas. El nodo hoja de árbol binario se denomina unidad de codificación (CU), que se usa para

la predicción (por ejemplo, predicción intraimagen o interimagen) y la transformación, sin ninguna partición adicional. Como se analizó anteriormente, las CU también pueden denominarse "bloques de vídeo" o "bloques".

En un ejemplo de la estructura de partición QTBT, el tamaño de CTU se establece como 128×128 (muestras de luma y dos muestras de croma correspondientes de 64×64), el MinQTSIZE se establece como 16×16 , el MaxBTSIZE se establece como 64×64 , el MinBTSIZE (tanto para el ancho como para la altura) se establece como 4, y el MaxBTDepth se establece como 4. La partición de árbol cuaternario se aplica a la CTU primero para generar nodos hoja de árbol cuaternario. Los nodos hoja del árbol cuaternario pueden tener un tamaño de 16×16 (es decir, el MinQTSIZE) a 128×128 (es decir, el tamaño de CTU). Si el nodo hoja del árbol cuaternario es de 128×128 , el árbol binario no lo dividirá más, ya que el tamaño excede el MaxBTSIZE (es decir, 64×64 , en este ejemplo). De cualquier otra manera, el nodo de árbol cuaternario de hoja se dividirá adicionalmente mediante el árbol binario. Por lo tanto, el nodo hoja del árbol cuaternario también es el nodo raíz del árbol binario y tiene la profundidad del árbol binario como 0. Cuando la profundidad del árbol binario alcanza MaxBTDepth (4, en este ejemplo), no se permite ninguna división adicional. Cuando el nodo del árbol binario tiene un ancho igual a MinBTSIZE (4, en este ejemplo), implica que no se permite ninguna división vertical adicional. De manera similar, un nodo de árbol binario que tiene una altura igual a MinBTSIZE implica que no se permite ninguna división horizontal adicional para ese nodo de árbol binario. Como se indicó anteriormente, los nodos hoja del árbol binario se denominan CU y se procesan además de acuerdo con la predicción y la transformación sin más partición.

La Figura 3 es un diagrama de flujo que ilustra un proceso ilustrativo para codificar información de movimiento mediante el uso de predictores de vectores de movimiento históricos (HMVP). Inicialmente, un codificador de vídeo, tal como el codificador de vídeo 200 o el decodificador de vídeo 300, carga una tabla con candidatos HMVP (140). El codificador de vídeo codifica entonces un bloque de datos de vídeo mediante el uso de los candidatos HMVP (142). El codificador de vídeo actualiza la tabla con la información de movimiento del bloque codificado (144).

La Figura 4 es un diagrama conceptual que ilustra un ejemplo de actualización de una tabla HMVP. En JVET-K0104, el tamaño de la tabla se establece en 16 y se aplica una regla de Primero en Primero (FIFO). La Figura 4 representa un ejemplo en el que la regla FIFO se aplica para eliminar un candidato HMVP y añadir uno nuevo a la tabla usada en los ejemplos de las técnicas de esta descripción.

Un codificador de vídeo, tal como el codificador de vídeo 200 o el decodificador de vídeo 300, puede insertar candidatos HMVP desde la última entrada hasta la primera entrada en la tabla después de un candidato de predicción de vector de movimiento temporal (TMVP) en una lista de candidatos. El codificador de vídeo puede aplicar la poda a los candidatos HMVP. El codificador de vídeo puede terminar el proceso de poda cuando el número total de candidatos de fusión disponibles alcanza un número máximo permitido señalado de candidatos de fusión.

En el ejemplo de la Figura 4, la tabla antes de la actualización incluye MVP_0 de historial ($HMVP_0$) a MVP_{L-1} de historial ($HMVP_{L-1}$), donde los números de subíndice 0 a $L-1$ representan un orden en el que se agregan los MVP históricos. C_{L-1} representa un nuevo MVP histórico a añadir a la tabla. Por lo tanto, de acuerdo con la regla FIFO, $HMVP_0$ se elimina de la tabla antes de añadir C_{L-1} , en este ejemplo.

La Figura 5 es un diagrama conceptual que ilustra una selección ilustrativa de bloques no adyacentes para la codificación de información de movimiento. En el ejemplo de la Figura 5, un bloque de corriente etiquetado "Corriente" representa una unidad de codificación (CU) de corriente, para la cual la información de movimiento puede codificarse mediante el uso de bloques vecinos adyacentes y/o no adyacentes etiquetados A_i , B_j , y NA_k . La predicción del vector de movimiento no adyacente se describe, por ejemplo, en Solicitud de los Estados Unidos núm. 16/003,269, presentada el 8 de junio de 2018. Un codificador de vídeo puede aplicar una regla FIFO y un tamaño máximo de memoria intermedia de candidatos de movimiento para bloques no adyacentes.

La Figura 6 es un diagrama conceptual que ilustra una selección de ejemplo de bloques no adyacentes en base a un bloque padre. Es decir, el bloque padre es un bloque dividido en subbloques, que incluye el bloque actual. Por ejemplo, el bloque padre puede ser una CTU o un subbloque en el que se ha dividido la CTU. Similar a la Figura 5, en la Figura 6, una CU actual se etiqueta como "Corriente", y los bloques no adyacentes de los cuales se puede recuperar información de movimiento y usar para predecir la información de movimiento de la CU actual se etiquetan como " $NA_{i,j}$ ".

Los vectores de movimiento de bloques espaciales adyacentes de un bloque colocalizado pueden usarse como candidatos de predicción de vectores de movimiento (MVP) para el modo de fusión, además de los vectores de movimiento H y C (es decir, vectores de movimiento en el centro y en la parte inferior derecha del bloque colocalizado).

Las técnicas de esta descripción pueden usarse para mejorar la predicción del vector de movimiento, por ejemplo, mediante la adición de candidatos usados para AMVP y/o modos de codificación de fusión, donde los candidatos añadidos pueden tomarse de bloques no adyacentes. Por ejemplo, los candidatos añadidos pueden corresponder a cualquiera de $NA_{1,1}$ a $NA_{1,9}$ de la Figura 6.

La Figura 7 es un diagrama conceptual que ilustra un ejemplo de un procesamiento de frente de onda deseado de unidades de árbol de codificación (CTU). Como se muestra en la Figura 7, pueden asignarse varios subprocesos para procesar diferentes líneas de CTU. Es decir, un codificador de vídeo, tal como el codificador de vídeo 200 o el decodificador de vídeo 300, puede ejecutar múltiples subprocesos diferentes para, por ejemplo, el procesamiento paralelo de frente de onda (WPP) cuando se codifican diferentes líneas de CTU. En algunos ejemplos, ciertas probabilidades usadas para la codificación basada en contexto (por ejemplo, codificación CABAC) de la información de movimiento de bloques interpredichos no podrían determinarse si las probabilidades se determinarían a partir de un último bloque de una línea CTU anterior, por ejemplo, al asumir que el último bloque aún no se había codificado. Por lo tanto, de acuerdo con las técnicas de esta descripción, el codificador de vídeo 200 y el decodificador de vídeo 300 restablecen una memoria intermedia de CTU para una línea de CTU antes de codificar la línea de CTU, para garantizar que las líneas de CTU puedan procesarse correctamente.

La Figura 8 es un diagrama conceptual que ilustra un ejemplo de información de movimiento usada para HMVP. La Figura 8 ilustra cómo el uso de FIFO puede eliminar vectores de movimiento de bloques más cercanos a un bloque actual de una lista de candidatos, mientras que podrían considerarse vectores de movimiento de bloques más lejanos. En particular, en la Figura 8, X representa la información de movimiento que se codifica actualmente, y los MV de los bloques sombreados están en la memoria intermedia de historial. Esta descripción reconoce que las técnicas HMVP convencionales no utilizan completamente los vectores de movimiento de bloques no adyacentes al menos en parte debido al uso de la regla FIFO, como se muestra en la Figura 8.

En particular, cuando se codifica el bloque X, la información de movimiento de bloques no adyacentes (TL0, T0, T1, TR0, TR1, TR2, TR3) de la CTU superior izquierda, CTU superior y CTU superior derecha se han eliminado de la memoria intermedia. Por lo tanto, la información de movimiento de estos bloques no se considera para su adición a la lista de candidatos, incluso si los bloques no adyacentes están más cerca de X que, por ejemplo, CTU LL0, CTU LL1 y CTU F0-F3, cuyos vectores de movimiento están en la memoria intermedia de historial.

Esta descripción también reconoce que una única memoria intermedia para HMVP no es aplicable para el procesamiento paralelo de frente de onda. Si solo se usa una única memoria intermedia, el tamaño de la memoria intermedia sería muy alto, para contener los candidatos espaciales potenciales para los bloques que se procesan en cada hilo (por ejemplo, línea CTU). Por ejemplo, si cuatro subprocesos se configuran para ejecutarse en paralelo, el tamaño de la memoria intermedia puede alcanzar 64. Como resultado, se necesitan más bits para señalar un índice de MVP al decodificador de vídeo 300. Igualmente, pueden ocurrir entradas redundantes. Es decir, una entrada en la memoria intermedia de historial puede ser potencialmente útil para un bloque en esta línea, pero puede no ser útil para bloques en otras líneas (por ejemplo, X y F en la Figura 8). En consecuencia, puede ser difícil encontrar el candidato óptimo para un bloque.

La Figura 9 es un diagrama conceptual que ilustra un ejemplo de una imagen dividida en múltiples líneas de unidades de árbol de codificación (CTU). En particular, en el ejemplo de la Figura 9, la imagen 150 incluye las líneas CTU 152A-152E (líneas CTU 152). Cada una de las líneas de CTU 152 incluye un conjunto respectivo de CTU: La línea CTU 152A incluye las CTU 154A-154J, la línea CTU 152B incluye las CTU 156A-156J, la línea CTU 152C incluye las CTU 158A-158J, la línea CTU 152D incluye las CTU 160A-160J y la línea CTU 152E incluye las CTU 162A-162J.

El codificador de vídeo 200 y el decodificador de vídeo 300 pueden configurarse de acuerdo con las técnicas de esta descripción para usar múltiples memorias intermedias para MVP basado en el historial. En algunos ejemplos, el codificador de vídeo 200 y el decodificador de vídeo 300 pueden mantener memorias intermedias de MVP histórico separados para cada una de las líneas de CTU 152 (cada una de las cuales puede procesarse mediante un hilo de procesamiento respectivo separado), o puede haber una única memoria intermedia que se restablece al comienzo de cada línea de CTU cuando se aplica el procesamiento paralelo de frente de onda.

En un ejemplo, la CTU 158C puede representar una CTU actual. La información de movimiento de las CTU 154A-154F, 156A-156D, 158A y 158B (mostrada mediante el uso de sombreado gris en la Figura 9) puede estar disponible en uno o más memorias intermedias de MVP histórico respectivos para su uso cuando se codifica la información de movimiento de la CTU 158C.

Adicional o alternativamente, el codificador de vídeo 200 y el decodificador de vídeo 300 pueden realizar la inicialización de una memoria intermedia de MVP histórico mediante el uso de cualquiera o todas las siguientes técnicas, solas o en combinación. El codificador de vídeo 200 y el decodificador de vídeo 300 restablecen la memoria intermedia de MVP histórico de cada línea CTU para que esté vacía. El codificador de vídeo 200 y el decodificador de vídeo 300 pueden rellenar previamente la memoria intermedia de MVP histórico de cada línea CTU con vectores de movimiento cero con un índice de trama de referencia diferente y/o direcciones de interpredicción, u otra información de movimiento predefinida o derivada. El codificador de vídeo 200 y el decodificador de vídeo 300 pueden rellenar previamente la memoria intermedia de MVP histórico de cada línea CTU con información de movimiento de las tramas codificadas (imágenes) en la misma capa temporal o capas temporales inferiores (imágenes de referencia que están disponibles para la trama/imágenes actuales).

5 El codificador de vídeo 200 y el decodificador de vídeo 300 pueden escalar la información de movimiento, por ejemplo, en base a la distancia temporal, o procesar/modificar la información de movimiento, por ejemplo, combinar esta información de movimiento con otro MV. En general, el codificador de vídeo 200 y el decodificador de vídeo 300 pueden combinar la información de movimiento con la información de movimiento de la memoria intermedia de MVP del historial anterior en las tramas/imágenes codificadas, o la información de movimiento de la región colocalizada (puede ser CTU, o mayor que un tamaño de bloque determinado, por ejemplo, bloque 4x4) en las tramas/imágenes codificadas. El codificador de vídeo 200 y el decodificador de vídeo 300 pueden rellenar previamente la memoria intermedia de MVP histórico de una línea CTU anterior cuando la CTU superior derecha de la CTU actual se codifica. El codificador de vídeo 200 y el decodificador de vídeo 300 pueden usar vectores de movimiento cero con diferentes índices de tramas de referencia y/o direcciones de interpretación, u otra información de movimiento predefinida o derivada.

15 Además, o alternativamente, el codificador de vídeo 200 y el decodificador de vídeo 300 pueden, siempre que se codifique (codifique o decodifique) una CTU de líneas de CTU, usar la memoria intermedia de MVP histórico asociada para inicializar o modificar las memorias intermedias de historial de la línea de CTU por debajo de la línea de CTU actual.

20 Adicional o alternativamente, el codificador de vídeo 200 y el decodificador de vídeo 300 pueden aplicar una regla FIFO para eliminar una entrada de una memoria intermedia de MVP histórico asociada.

Adicional o alternativamente, el codificador de vídeo 200 y el decodificador de vídeo 300 pueden borrar una memoria intermedia de MVP histórico cuando la línea CTU asociada se codifica/decodifica completamente.

25 El codificador de vídeo 200 y el decodificador de vídeo 300 pueden mantener un tamaño de la memoria intermedia de MVP que es mayor que AMVP/fusión u otras listas de candidatos de modo. Cualquier MV o MV de la memoria intermedia puede seleccionarse como candidato(s) de MV para la lista de candidatos usada en un modo determinado, por ejemplo, modo de fusión AMPV, afin, o cualquier otro modo intermedio. Puede definirse una regla sobre cómo seleccionar los MV de la memoria intermedia, por ejemplo, tomar los últimos N MV añadidos a la memoria intermedia, o tomar algunos del principio de la memoria intermedia, y/o algunos del medio de la memoria intermedia, y/o algunos del final de la memoria intermedia. Alternativamente, la señalización puede aplicarse para indicar que MV se seleccionan (por ejemplo, el codificador de vídeo 200 puede codificar los datos señalizados, y el decodificador de vídeo 300 puede decodificar los datos señalizados). El tamaño de la memoria intermedia de MVP puede señalarse en cualquier conjunto de parámetros (por ejemplo, conjunto de parámetros de vídeo, conjunto de parámetros de secuencia, conjunto de parámetros de imagen o similares), encabezado de segmento o en otro lugar. La memoria intermedia de MVP puede asociarse con un segmento, imagen y/o secuencia de vídeo.

40 Cuando el codificador de vídeo 200 y el decodificador de vídeo 300 procesan un bloque intercodificado, los MV usados en el bloque pueden añadirse a la memoria intermedia de MVP, y solo los MV únicos pueden mantenerse en la memoria intermedia. Cuando el amortiguador está lleno, los MV más antiguos pueden eliminarse del amortiguador cuando se añade un nuevo MV. Puede haber una regla mediante la cual los MV pueden añadirse a la memoria intermedia, por ejemplo, solo el MV señalado, tal como en el modo AMVP, puede añadirse, y si un bloque se codifica en modo de fusión, el MV del bloque no se añade a la memoria intermedia.

45 El codificador de vídeo 200 y el decodificador de vídeo 300 pueden agregar un MV a un MV o MV ya existente en la memoria intermedia. Por ejemplo, si los MV existentes en la memoria intermedia son unidireccionales, cuando se añade un nuevo MV, esos MV existentes pueden modificarse para que sean bidireccionales mediante la adición del nuevo MV.

50 Puede aplicarse algún procesamiento de MV mientras se añade un nuevo MV. Por ejemplo, si un nuevo MV está cerca de los MV ya existentes en la memoria intermedia, esos MV cercanos pueden eliminarse. Cercano puede significar cercano al comparar los valores de los componentes MV (por ejemplo, componentes x e y). En algunos ejemplos, solo los MV que difieren de los MV ya existentes en el amortiguador por un umbral pueden añadirse al amortiguador. El mismo umbral puede configurarse para diferentes amortiguadores.

55 Los vectores de movimiento en la memoria intermedia pueden ser unidireccionales (L0 o L1), bidireccionales o cualquier otro modelo de movimiento MV.

60 La información de modo puede asociarse con un MV en la memoria intermedia, y si se señala un índice de un MV en la memoria intermedia en el bloque u otra regla se aplica sobre la obtención de MV de la memoria intermedia, entonces la información de modo puede derivarse de los datos asociados con esa información de MV. Por ejemplo, si esa información está en modo de fusión, entonces el bloque se codifica en modo de fusión con el MV indicado.

65 Esta descripción reconoce además que los MVP convencionales basados en el historial solo contienen predictores de movimiento regulares y solo se usan para la predicción del movimiento regular sin modificar la información de movimiento. De acuerdo con las técnicas de esta descripción, el codificador de vídeo 200 y el decodificador de vídeo 300 pueden usar al menos una memoria intermedia de MVP histórico que contiene no solo información de movimiento

codificada, sino también otros tipos de predictores de movimiento, tales como para el modelo de movimiento afín, información de movimiento de modo de copia intrabloque, información de movimiento de compensación de iluminación local, MVP de subbloque o predictor de movimiento temporal.

5 Adicional o alternativamente, el codificador de vídeo 200 y el decodificador de vídeo 300 pueden usar múltiples memorias intermedias de MVP histórico para diferentes modelos de movimiento, tales como modelo de movimiento afín, información de movimiento de modo de copia intrabloque, información de movimiento de compensación de iluminación local, MVP de subbloque o predictor de movimiento temporal.

10 Además, o alternativamente, el vector de movimiento sintético en base al MVP actual y otro predictor de movimiento, tal como otro MVP espacial, o MVP temporal, también puede añadirse a la lista de candidatos.

15 Adicional o alternativamente, el codificador de vídeo 200 y el decodificador de vídeo 300 pueden generar un MVP sintético a partir de dos o más MVP en la memoria intermedia de MVP histórico o uno o más MVP en la memoria intermedia de MVP histórico con otro tipo de MVP, tales como MVP espaciales o temporales.

20 El codificador de vídeo 200 y el decodificador de vídeo 300 pueden implementar un esquema de partición de bloques. En HEVC, las imágenes se dividen en una secuencia de unidades de árbol de codificación (CTU). Para una imagen que tiene tres matrices de muestra, una CTU incluye un bloque N×N de muestras de luma junto con dos bloques correspondientes de muestras de croma. Una CTU se divide en unidades de codificación (CU) mediante el uso de una estructura de árbol. Cada CU hoja puede dividirse además en una, dos o cuatro unidades de predicción (PU) de acuerdo con el tipo de división de PU. Después de obtener el bloque residual mediante la aplicación del proceso de predicción en base al tipo de división de PU, una CU hoja puede dividirse en unidades de transformación (TU).

25 En VVC, un árbol cuaternario con árbol de múltiples tipos anidados mediante el uso de una estructura de segmentación de divisiones binarias y ternarias reemplaza los conceptos de múltiples tipos de unidades de partición, es decir, la partición del árbol de múltiples tipos anidado elimina la separación de los conceptos de CU, PU y TU, excepto cuando sea necesario para las CU que tienen un tamaño demasiado grande para la longitud máxima de la transformada y admiten más flexibilidad para las formas de partición de CU. En la estructura del árbol de codificación, una CU puede tener una forma cuadrada o rectangular.

30 El codificador de vídeo 200 y el decodificador de vídeo 300 pueden utilizar información de movimiento para predecir un bloque de datos de vídeo. Para cada bloque, puede estar disponible un conjunto de información de movimiento. El conjunto de información de movimiento contiene información de movimiento para las direcciones de predicción hacia adelante y hacia atrás. Aquí, las direcciones de predicción hacia adelante y hacia atrás son dos direcciones de predicción correspondientes a la lista de imágenes de referencia 0 (RefPicList0) y la lista de imágenes de referencia 1 (RefPicList1) de una imagen o segmento actual. Los términos "hacia adelante" y "hacia atrás" no tienen necesariamente un significado geométrico. En cambio, se usan para distinguir en qué lista de imágenes de referencia se basa un vector de movimiento. La predicción hacia adelante significa la predicción formada en base a la lista de referencia 0, mientras que la predicción hacia atrás significa la predicción formada en base a la lista de referencia 1. En los casos en los que ambas listas de referencia 0 y lista de referencia 1 se usan para formar una predicción para un bloque dado, se llama predicción bidireccional.

45 Para una imagen o segmento dada, si solo se usa una lista de imágenes de referencia, cada bloque dentro de la imagen o segmento se predice hacia adelante. Si ambas listas de imágenes de referencia se usan para una imagen o segmento dada, un bloque dentro de la imagen o segmento puede predecirse hacia adelante, o predecirse hacia atrás, o predecirse bidireccionalmente.

50 Para cada dirección de predicción, la información de movimiento contiene un índice de referencia y un vector de movimiento. Se usa un índice de referencia para identificar una imagen de referencia en la lista de imágenes de referencia correspondiente (por ejemplo, RefPicList0 o RefPicList1). Un vector de movimiento tiene una componente horizontal y una vertical, cada una de las cuales indica un valor de desplazamiento a lo largo de la dirección horizontal y vertical respectivamente. En algunas descripciones, para simplificar, la palabra "vector de movimiento" puede usarse indistintamente con la información de movimiento, para indicar tanto el vector de movimiento como su índice de referencia asociado.

55 El recuento de orden de imágenes (POC) se usa ampliamente en los estándares de codificación de vídeo para identificar un orden de visualización de una imagen. Aunque hay casos en los que dos imágenes dentro de una secuencia de vídeo codificada pueden tener el mismo valor de POC, típicamente no sucede dentro de una secuencia de vídeo codificada. Cuando múltiples secuencias de vídeo codificadas están presentes en un flujo de bits, las imágenes con un mismo valor de POC pueden estar más cerca entre sí en términos de orden de decodificación.

60 En HEVC, hay dos modos de predicción inter, denominados modo de fusión (con el salto que se considera un caso especial del modo de fusión) y modo de predicción avanzada de vectores de movimiento (AMVP), respectivamente para una PU.

65

En modo AMVP o de fusión, se mantiene una lista de candidatos de vectores de movimiento (MV) para múltiples predictores de vectores de movimiento. Los vectores de movimiento, así como también los índices de referencia en el modo de fusión, de la PU actual se generan tomando un candidato de la lista de candidatos del MV.

5 La lista de candidatos MV contiene hasta 5 candidatos para el modo de fusión y solo dos candidatos para el modo AMVP. Un candidato de fusión puede contener un conjunto de información de movimiento, por ejemplo, vectores de movimiento correspondientes a ambas listas de imágenes de referencia (lista 0 y lista 1) y los índices de referencia. Si un candidato de fusión se identifica mediante un índice de fusión, las imágenes de referencia se usan para la predicción de los bloques actuales, así como también se determinan los vectores de movimiento asociados. Sin embargo, en el modo AMVP para cada dirección de predicción potencial de la lista 0 o la lista 1, se señala explícitamente un índice de referencia, junto con un índice MVP a la lista de candidatos MV ya que el candidato AMVP contiene solo un vector de movimiento. En el modo de AMVP, los vectores de movimiento previstos pueden refinarse aún más.

15 Como puede verse arriba, un candidato de fusión corresponde a un conjunto completo de información de movimiento mientras que un candidato de AMVP contiene solo un vector de movimiento para una dirección de predicción e índice de referencia específicos. Los candidatos para ambos modos pueden derivarse de manera similar de los mismos bloques vecinos espaciales y temporales.

20 Las Figuras 10A y 10B son diagramas de bloques que ilustran candidatos de vectores de movimiento vecinos espaciales ilustrativos para los modos de fusión y predicción avanzada de vectores de movimiento (AMVP). La Figura 10A muestra un ejemplo de candidatos MV vecinos espaciales para el modo de fusión, y la Figura 10B muestra un ejemplo de candidatos MV vecinos espaciales para el modo AMVP. Los candidatos de MV espaciales se derivan de los bloques vecinos como se muestra en las Figuras 10A y 10B. Para un PU específico (PU0), los procedimientos para generar los candidatos a partir de los bloques difieren para los modos de fusión y AMVP.

25 En el modo de fusión, pueden derivarse hasta cuatro candidatos de MV espaciales con el orden que se muestra en la Figura 10A. Específicamente, el orden es el siguiente: izquierda (0), arriba (1), arriba derecha (2), abajo izquierda (3) y arriba izquierda (4), como se muestra en la Figura 10A.

30 En el modo AVMP, los bloques vecinos se dividen en dos grupos. El primer grupo es un grupo izquierdo que incluye los bloques 0 y 1. El segundo grupo es un grupo superior que incluye los bloques 2, 3 y 4, como se muestra en la Figura 10B. Para cada grupo, el candidato potencial en un bloque vecino que se refiere a la misma imagen de referencia que la indicada por el índice de referencia señalado tiene la prioridad más alta para ser elegido para formar un candidato final del grupo. Es posible que todos los bloques vecinos no contengan un vector de movimiento que apunte a la misma imagen de referencia. Por lo tanto, si no se puede encontrar tal candidato, el primer candidato disponible se escalará para formar el candidato final, por lo que las diferencias de distancia temporal pueden compensarse.

35 40 Las Figuras 11A y 11B son diagramas conceptuales que ilustran los candidatos de predicción del vector de movimiento temporal (TMVP). La Figura 11A muestra un ejemplo de un candidato TMVP. Un candidato TMVP, si está habilitado y disponible, se añade a la lista de candidatos MV después de los candidatos de vectores de movimiento espacial. El proceso de derivación del vector de movimiento para un candidato TMVP es el mismo tanto para el modo de fusión como para el AMVP, sin embargo, el índice de referencia objetivo para el candidato TMVP en el modo de fusión siempre se establece en 0.

45 La ubicación del bloque primario para la derivación de candidatos TMVP es el bloque inferior derecho fuera de la PU colocalizada como se muestra en la Figura 11A como un bloque "T", para compensar el sesgo hacia los bloques superior e izquierdo usados para generar candidatos vecinos espaciales. Sin embargo, si ese bloque se ubica fuera de la hilera CTB actual o la información de movimiento no está disponible, el bloque se sustituye con un bloque central de la PU.

50 Un vector de movimiento para un candidato TMVP se deriva de la PU colocalizada de la imagen colocalizada, indicada en el nivel de segmento. El vector de movimiento para la PU colocalizada se denomina MV colocalizado.

55 La Figura 11B muestra un ejemplo de escalado de MV. Para derivar el vector de movimiento candidato TMVP, el MV colocalizado puede necesitar escalarse para compensar las diferencias de distancia temporal, como se muestra en la Figura 11B.

60 Varios otros aspectos de los modos de fusión y AMVP merecen mencionarse como sigue. Por ejemplo, el codificador de vídeo 200 y el decodificador de vídeo 300 pueden realizar el escalado de vectores de movimiento. Se supone que el valor de los vectores de movimiento es proporcional a la distancia de las imágenes en el tiempo de presentación. Un vector de movimiento asocia dos imágenes, la imagen de referencia y la imagen que contiene el vector de movimiento (es decir, la imagen que contiene). Cuando se utiliza un vector de movimiento para predecir el otro vector de movimiento, la distancia de la imagen que contiene y la imagen de referencia se calcula en base a los valores de recuento de orden de imágenes (POC).

Para predecir un vector de movimiento, tanto su imagen contenedora asociada como la imagen de referencia pueden ser diferentes. Por lo tanto, se calcula una nueva distancia (basada en POC). Y el vector de movimiento se escala en base a estas dos distancias POC. Para un candidato vecino espacial, las imágenes que contienen para los dos vectores de movimiento son las mismas, mientras que las imágenes de referencia son diferentes. En HEVC, el escalado del vector de movimiento se aplica tanto a TMVP como a AMVP para candidatos vecinos espaciales y temporales.

Como otro ejemplo, el codificador de vídeo 200 y el decodificador de vídeo 300 pueden realizar la generación de candidatos de vectores de movimiento artificiales. Si una lista de candidatos de vectores de movimiento no está completa, se generan candidatos de vectores de movimiento artificiales y se insertan al final de la lista hasta que tenga todos los candidatos.

En el modo de fusión, hay dos tipos de candidatos de MV artificiales: candidatos combinados derivados solo para segmentos B y candidatos cero usados solo para AMVP si el primer tipo no proporciona suficientes candidatos artificiales. Para cada par de candidatos que ya están en la lista de candidatos y tienen la información de movimiento necesaria, los candidatos de vectores de movimiento combinados bidireccionales se derivan mediante una combinación del vector de movimiento del primer candidato que se refiere a una imagen en la lista 0 y el vector de movimiento de un segundo candidato que se refiere a una imagen en la lista 1.

Como otro ejemplo, el codificador de vídeo 200 y el decodificador de vídeo 300 pueden realizar un proceso de poda para la inserción de candidatos. Los candidatos de diferentes bloques pueden resultar ser los mismos, lo que disminuye la eficiencia de una lista de candidatos de fusión/AMVP. Se aplica un proceso de poda para resolver este problema. Compara un candidato con los otros de la lista de candidatos actual para evitar insertar un candidato idéntico en cierta medida. Para reducir la complejidad, solo se aplica un número limitado de procesos de poda en lugar de comparar cada uno potencial con todos los demás existentes. Solo se aplican las siguientes comparaciones si corresponde: el candidato de fusión superior se compara con el candidato de fusión izquierdo, el candidato de fusión derecho superior se compara con el candidato de fusión superior, el candidato de fusión inferior izquierdo se compara con el candidato de fusión izquierdo, el candidato de fusión superior izquierdo se compara con el candidato de fusión izquierdo y el candidato de fusión superior.

El codificador de vídeo 200 y el decodificador de vídeo 300 también pueden usar otros procedimientos de predicción de movimiento. En el desarrollo de la Codificación de Vídeo Versátil (VVC), se propuso un procedimiento de predicción de vectores de movimiento basado en el historial (HMVP) en L. Zhang etc., "CE4-related: History-based Motion Vector Prediction", Joint Video Experts Team Document: JVET-K0104 (en adelante "K0104"). El procedimiento HMVP permite que cada bloque encuentre su predictor MV a partir de una lista de MV decodificados del pasado además de los campos de movimiento vecinos causales adyacentes inmediatos. Se mantiene una tabla con múltiples candidatos de HMVP durante el proceso de codificación/decodificación. La tabla se vacía cuando se encuentra un nuevo segmento. Siempre que haya un bloque intercodificado, la información de movimiento asociada se inserta en la tabla de una manera de primero en entrar primero en salir (FIFO) como un nuevo candidato HMVP. Después, puede aplicarse una regla FIFO de restricción. Cuando se inserta un HMVP en la tabla, se puede aplicar primero una comprobación de redundancia para averiguar si hay un HMVP idéntico en la tabla. Si se encuentra, ese HMVP particular puede eliminarse de la tabla y todos los candidatos HMVP posteriores se mueven.

Los candidatos HMVP pueden usarse en el proceso de construcción de la lista de candidatos a la fusión. Por ejemplo, todos los candidatos HMVP de la última entrada a la primera entrada de la tabla pueden insertarse después del candidato TMVP. La poda puede aplicarse a los candidatos HMVP. Una vez que el número total de candidatos de fusión disponibles alcanza los candidatos de fusión máximamente permitidos señalados, el proceso de construcción de la lista de candidatos de fusión se termina.

De manera similar, los candidatos HMVP también pueden usarse en el proceso de construcción de la lista de candidatos AMVP. Los vectores de movimiento de los últimos K candidatos HMVP en la tabla pueden insertarse después del candidato TMVP. En algunos ejemplos, solo los candidatos HMVP con la misma imagen de referencia que la imagen de referencia de destino AMVP se usan para construir la lista de candidatos AMVP. La poda puede aplicarse a los candidatos HMVP.

En HEVC, la codificación de una CTU actual solo puede depender de las CTU a la izquierda, arriba-izquierda, arriba y arriba-derecha. Por lo tanto, el procesamiento paralelo de frente de onda (WPP) puede admitirse en HEVC. Sin embargo, el procedimiento HMVP en K0104 puede provocar dependencia entre un bloque actual y todas las CTU codificadas previamente en un segmento. Por lo tanto, WPP puede no aplicarse si se usa el procedimiento HMVP. Esta descripción describe técnicas para usar HMVP con inicialización CTU, en las que las dependencias permanecen iguales que en HEVC. Esta descripción también describe técnicas para HMVP con inicialización de hilera CTU (reinicio).

De acuerdo con las técnicas de esta descripción, el codificador de vídeo 200 y el decodificador de vídeo 300 pueden realizar HMVP con inicialización de CTU. La tabla HMVP se inicializa al comienzo de cada CTU. La inicialización puede

añadir MV de los bloques codificados vecinos inmediatos de la CTU actual a la tabla HMVP. El bloque codificado vecino inmediato puede estar a la izquierda, arriba, arriba-izquierda o arriba-derecha de la CTU actual como en HEVC. Si se habilita la predicción del vector de movimiento temporal, el bloque codificado vecino inmediato también puede ser el bloque de referencia en la imagen de referencia.

5 La Figura 12 es un diagrama de bloques que ilustra un ejemplo de una unidad de árbol de codificación (CTU) y bloques vecinos. En un ejemplo, solo los candidatos de fusión espacial y temporal para el bloque CTU actual se usan para inicializar la tabla HMVP. Un ejemplo de uso de candidatos de fusión temporal y espacial HEVC se muestra en la Figura 12. El orden de inserción es el siguiente: izquierda (0), arriba (1), arriba derecha (2) y arriba izquierda (4). La ubicación del candidato de fusión temporal se indica mediante "T". Tenga en cuenta que el candidato de fusión temporal inferior derecha y el candidato inferior izquierdo (3) no están disponibles ya que sus ubicaciones están por debajo de la línea CTU actual.

10 En otro ejemplo, el proceso de derivación de candidatos de fusión para el bloque CTU actual se usa para inicializar la tabla HMVP. Además de los candidatos de fusión espaciales y temporales, también pueden usarse otros candidatos de fusión (candidatos de vectores de movimiento artificiales, por ejemplo) para la inicialización.

15 La Figura 13 es un diagrama de bloques que ilustra una CU actual en una CTU actual. En algunos ejemplos, la tabla HMVP se inicializa como vacía al comienzo de la codificación de la CTU actual. Sin embargo, después de que se codifica la primera CU, los candidatos de fusión espacial y temporal de la primera CU, como se muestra en la Figura 13, se añaden a la tabla HMVP. Y después, también se añade el MV de la primera CU si la CU se codifica por interpredicción. Tenga en cuenta que, si la primera CU no es igual a la CTU actual, pueden añadirse dos candidatos de fusión temporal "T0" y "T1" en orden. La Figura 13 muestra un ejemplo de candidatos de fusión de la primera CU en una CTU.

20 En otro ejemplo, la tabla HMVP se inicializa como vacía al comienzo de la codificación de la CTU actual. Sin embargo, después de codificar la primera CU, todos los candidatos de fusión de la primera CU se añaden a la tabla HMVP. Y después también se añade el MV de la primera CU si se codifica por interpredicción.

25 El codificador de vídeo 200 y el decodificador de vídeo 300 también pueden realizar HMVP con inicialización de hilera CTU. En otro ejemplo, la inicialización de CTU para HMVP descrita anteriormente se aplica solo a la primera CTU en una hilera de CTU. De manera similar al HMVP en K0104, el proceso de poda puede aplicarse a la tabla inicializada para eliminar algunos o todos los duplicados. El proceso de poda tampoco puede aplicarse a la tabla inicializada para reducir la complejidad.

30 La Figura 14 es un diagrama de bloques que ilustra un codificador de vídeo 200 ilustrativo que puede realizar las técnicas de esta descripción. La Figura 14 se proporciona con fines de explicación y no debe considerarse limitante de las técnicas como se ejemplifica y describe ampliamente en esta descripción. Para fines de explicación, esta descripción describe el codificador de vídeo 200 en el contexto de los estándares de codificación de vídeo, tales como el estándar de codificación de vídeo HEVC y el estándar de codificación de vídeo H.266 en desarrollo. Sin embargo, las técnicas de esta descripción no se limitan a estos estándares de codificación de vídeo y son aplicables generalmente a la codificación y decodificación de vídeo.

35 En el ejemplo de la Figura 14, el codificador de vídeo 200 incluye la memoria de datos de vídeo 230, la unidad de selección de modo 202, la unidad de generación residual 204, la unidad de procesamiento de transformación 206, la unidad de cuantificación 208, la unidad de cuantificación inversa 210, la unidad de procesamiento de transformación inversa 212, la unidad de reconstrucción 214, la unidad de filtro 216, la memoria intermedia de imágenes decodificadas (DPB) 218 y la unidad de codificación por entropía 220. Cualquiera o todas las memorias de datos de vídeo 230, la unidad de selección de modo 202, la unidad de generación residual 204, la unidad de procesamiento de transformación 206, la unidad de cuantificación 208, la unidad de cuantificación inversa 210, la unidad de procesamiento de transformación inversa 212, la unidad de reconstrucción 214, la unidad de filtro 216, el DPB 218 y la unidad de codificación de entropía 220 pueden implementarse en uno o más procesadores o en circuitos de procesamiento. Además, el codificador de vídeo 200 puede incluir procesadores o circuitos de procesamiento adicionales o alternativos para realizar estas y otras funciones.

40 La memoria de datos de vídeo 230 puede almacenar los datos de vídeo para ser codificados por los componentes del codificador de vídeo 200. El codificador de vídeo 200 puede recibir los datos de vídeo almacenados en la memoria de datos de vídeo 230 de, por ejemplo, la fuente de vídeo 104 (Figura 1). El DPB 218 puede actuar como una memoria de imágenes de referencia que almacena datos de vídeo de referencia para su uso en la predicción de datos de vídeo posteriores por el codificador de vídeo 200. La memoria de datos de vídeo 230 y la DPB 218 pueden formarse por cualquiera de una variedad de dispositivos de memoria, tal como la memoria dinámica de acceso aleatorio (DRAM), incluida la DRAM síncrona (SDRAM), la RAM magnetorresistiva (MRAM), la RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. La memoria de datos de vídeo 230 y la DPB 218 pueden proporcionarse por el mismo dispositivo de memoria o por dispositivos de memoria separados. En varios ejemplos, la memoria de datos de vídeo 230 puede estar en el chip con otros componentes del codificador de vídeo 200, como se ilustra, o fuera del chip con relación a esos componentes.

En esta descripción, la referencia a la memoria de datos de vídeo 230 no debe interpretarse como que se limita a la memoria interna al codificador de vídeo 200, a menos que se describa específicamente como tal, o la memoria externa al codificador de vídeo 200, a menos que se describa específicamente como tal. Más bien, la referencia a la memoria de datos de vídeo 230 debe entenderse como una memoria de referencia que almacena datos de vídeo que el codificador de vídeo 200 recibe para codificar (por ejemplo, datos de vídeo para un bloque actual que se va a codificar). La memoria 106 de la Figura 1 también puede proporcionar almacenamiento temporal de las salidas de las diversas unidades del codificador de vídeo 200.

Las diversas unidades de la Figura 14 se ilustran para ayudar a comprender las operaciones realizadas por el codificador de vídeo 200. Las unidades pueden implementarse como circuitos de función fija, circuitos programables, o una de sus combinaciones. Los circuitos de función fija se refieren a circuitos que proporcionan una funcionalidad particular y se preestablecen en las operaciones que pueden realizarse. Los circuitos programables se refieren a circuitos que pueden programarse para realizar diversas tareas y proporcionar funcionalidad flexible en las operaciones que pueden realizarse. Por ejemplo, los circuitos programables pueden ejecutar software o microprograma que hacen que los circuitos programables funcionen de la manera definida por las instrucciones del software o microprograma. Los circuitos de función fija pueden ejecutar instrucciones de software (por ejemplo, para recibir parámetros o parámetros de salida), pero los tipos de operaciones que realizan los circuitos de función fija son generalmente inmutables. En algunos ejemplos, una o más de las unidades pueden ser bloques de circuito distintos (de función fija o programable), y en algunos ejemplos, una o más unidades pueden ser circuitos integrados.

El codificador de vídeo 200 puede incluir unidades lógicas aritméticas (ALU), unidades de función elementales (EFU), circuitos digitales, circuitos analógicos y/o núcleos programables, formados a partir de circuitos programables. En ejemplos donde las operaciones del codificador de vídeo 200 se realizan mediante el uso de software ejecutado por los circuitos programables, la memoria 106 (Figura 1) puede almacenar el código de objeto del software que recibe y ejecuta el codificador de vídeo 200, u otra memoria dentro del codificador de vídeo 200 (no mostrada) puede almacenar tales instrucciones.

La memoria de datos de vídeo 230 se configura para almacenar los datos de vídeo recibidos. El codificador de vídeo 200 puede recuperar una imagen de los datos de vídeo de la memoria de datos de vídeo 230 y proporcionar los datos de vídeo a la unidad de generación residual 204 y a la unidad de selección de modo 202. Los datos de vídeo en la memoria de datos de vídeo 230 pueden ser datos de vídeo sin procesar que se van a codificar.

La unidad de selección de modo 202 incluye una unidad de estimación de movimiento 222, una unidad de compensación de movimiento 224 y una unidad de intrapredicción 226. La unidad de selección de modo 202 puede incluir unidades funcionales adicionales para realizar la predicción de vídeo de acuerdo con otros modos de predicción. Como ejemplos, la unidad de selección de modo 202 puede incluir una unidad de paleta, una unidad de copia intrabloque (que puede ser parte de la unidad de estimación de movimiento 222 y/o la unidad de compensación de movimiento 224), una unidad afín, una unidad de modelo lineal (LM), o similares.

La unidad de selección de modo 202 generalmente coordina múltiples pasadas de codificación para probar combinaciones de parámetros de codificación y valores de velocidad de distorsión resultantes para tales combinaciones. Los parámetros de codificación pueden incluir la partición de las CTU en las CU, los modos de predicción para las CU, los tipos de transformación para los datos residuales de las CU, los parámetros de cuantificación para los datos residuales de las CU, etc. La unidad de selección de modo 202 puede seleccionar finalmente la combinación de parámetros de codificación que tienen valores de velocidad de distorsión que son mejores que las otras combinaciones probadas.

El codificador de vídeo 200 puede dividir una imagen recuperada de la memoria de datos de vídeo 230 en una serie de CTU y encapsular una o más CTU dentro de un segmento. La unidad de selección de modo 202 puede dividir una CTU de la imagen de acuerdo con una estructura de árbol, tal como la estructura QTBT o la estructura de árbol cuádruple de HEVC descrita anteriormente. Como se describió anteriormente, el codificador de vídeo 200 puede formar una o más CU a partir de la partición de una CTU de acuerdo con la estructura del árbol. Tal CU también puede denominarse generalmente "bloque de vídeo" o "bloque".

En general, la unidad de selección de modo 202 también controla los componentes de esta (por ejemplo, la unidad de estimación de movimiento 222, la unidad de compensación de movimiento 224 y la unidad de intrapredicción 226) para generar un bloque de predicción para un bloque actual (por ejemplo, una CU actual, o en HEVC, la porción superpuesta de una PU y una TU). Para la interpredicción de un bloque actual, la unidad de estimación de movimiento 222 puede realizar una búsqueda de movimiento para identificar uno o más bloques de referencia que coinciden estrechamente en una o más imágenes de referencia (por ejemplo, una o más imágenes codificadas previamente almacenadas en DPB 218). En particular, la unidad de estimación de movimiento 222 puede calcular un valor representativo de cuán similar es un bloque de referencia potencial al bloque actual, por ejemplo, de acuerdo con la suma de la diferencia absoluta (SAD), la suma de las diferencias al cuadrado (SSD), la diferencia absoluta media (MAD), las diferencias al cuadrado medias (MSD), o similares. La unidad de estimación de movimiento 222 generalmente puede realizar estos cálculos mediante el uso de diferencias muestra por muestra entre el bloque actual y el bloque de referencia que se considera. La unidad de estimación de movimiento 222 puede identificar un bloque

de referencia que tiene un valor más bajo resultante de estos cálculos, lo que indica un bloque de referencia que coincide más estrechamente con el bloque actual.

5 La unidad de estimación de movimiento 222 puede formar uno o más vectores de movimiento (MV) que definen las posiciones de los bloques de referencia en las imágenes de referencia con relación a la posición del bloque actual en una imagen actual. La unidad de estimación de movimiento 222 puede entonces proporcionar los vectores de movimiento a la unidad de compensación de movimiento 224. Por ejemplo, para la interpredicción unidireccional, la unidad de estimación de movimiento 222 puede proporcionar un solo vector de movimiento, mientras que para la interpredicción bidireccional, la unidad de estimación de movimiento 222 puede proporcionar dos vectores de movimiento. La unidad de compensación de movimiento 224 puede generar un bloque de predicción mediante el uso de los vectores de movimiento. Por ejemplo, la unidad de compensación de movimiento 224 puede recuperar datos del bloque de referencia mediante el uso del vector de movimiento. Como otro ejemplo, si el vector de movimiento tiene una precisión de muestra fraccionada, la unidad de compensación de movimiento 224 puede interpolar valores para el bloque de predicción de acuerdo con uno o más filtros de interpolación. Además, para la interpredicción bidireccional, la unidad de compensación de movimiento 224 puede recuperar datos para dos bloques de referencia identificados por vectores de movimiento respectivos y combinar los datos recuperados, por ejemplo, mediante el promedio de muestra a muestra o el promedio ponderado.

20 De acuerdo con las técnicas de esta descripción, la memoria intermedia de imágenes decodificadas 218 puede incluir una o más memorias intermedias de MVP histórico para líneas de CTU. Es decir, a cada línea CTU se le puede asignar su propia memoria intermedia de MVP, o se puede usar una única memoria intermedia MVP para una pluralidad de líneas CTU. En cualquier caso, el codificador de vídeo 200 puede restablecer una memoria intermedia de MVP para una línea CTU al comienzo de la decodificación de datos de vídeo de la línea CTU. La unidad de compensación de movimiento 224 u otra unidad del codificador de vídeo 200 puede configurarse para almacenar solo vectores de movimiento únicos en la memoria intermedia de MVP. Como se analizó anteriormente, la unidad de compensación de movimiento 224 u otra unidad del codificador de vídeo 200 puede configurarse para usar una regla FIFO para gestionar la información de movimiento almacenada en la memoria intermedia de MVP, de manera que cuando se añade un vector de movimiento a la memoria intermedia de MVP, si la memoria intermedia de MVP está llena, la unidad de compensación de movimiento 224 puede eliminar un vector de movimiento añadido más temprano de la memoria intermedia de MVP. En algunos ejemplos, el codificador de vídeo 200 puede mantener diferentes memorias intermedias de MVP respectivos para cada uno de una variedad de modelos de movimiento, tales como, por ejemplo, modelo de movimiento afín, información de movimiento de modo de copia intrabloque, información de movimiento de compensación de iluminación local, MVP de subbloque y predicción de movimiento temporal.

35 Como otro ejemplo, para la intrapredicción, o la codificación de intrapredicción, la unidad de intrapredicción 226 puede generar el bloque de predicción a partir de muestras vecinas al bloque actual. Por ejemplo, para los modos direccionales, la unidad de intrapredicción 226 puede combinar generalmente de manera matemática los valores de las muestras vecinas y poblar estos valores calculados en la dirección definida a través del bloque actual para producir el bloque de predicción. Como otro ejemplo, para el modo DC, la unidad de predicción interna 226 puede calcular un promedio de las muestras vecinas al bloque actual y generar el bloque de predicción para incluir este promedio resultante para cada muestra del bloque de predicción.

45 La unidad de selección de modo 202 proporciona el bloque de predicción a la unidad de generación residual 204. La unidad de generación residual 204 recibe una versión sin codificar y sin procesar del bloque actual de la memoria de datos de vídeo 230 y el bloque de predicción de la unidad de selección de modo 202. La unidad de generación residual 204 calcula las diferencias muestra a muestra entre el bloque actual y el bloque de predicción. Las diferencias de muestra a muestra resultantes definen un bloque residual para el bloque actual. En algunos ejemplos, la unidad de generación residual 204 también puede determinar diferencias entre los valores de muestra en el bloque residual para generar un bloque residual mediante el uso de modulación de código de pulso diferencial residual (RDPCM). En algunos ejemplos, la unidad de generación residual 204 puede formarse mediante el uso de uno o más circuitos sustractores que realizan la sustracción binaria.

55 En ejemplos donde la unidad de selección de modo 202 divide las CU en PU, cada PU puede asociarse con una unidad de predicción de luma y unidades de predicción de croma correspondientes. El codificador de vídeo 200 y el decodificador de vídeo 300 pueden soportar las PU que tienen diversos tamaños. Como se indicó anteriormente, el tamaño de una CU puede referirse al tamaño del bloque de codificación de luma de la CU y el tamaño de una PU puede referirse al tamaño de una unidad de predicción de luma de la PU. Suponiendo que el tamaño de una CU particular es $2N \times 2N$, el codificador de vídeo 200 puede admitir tamaños de PU de $2N \times 2N$ o $N \times N$ para la intrapredicción, y tamaños de PU simétricos de $2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$ o similares para la interpredicción. El codificador de vídeo 200 y el decodificador de vídeo 300 también pueden soportar particiones asimétricas para tamaños de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$ y $nR \times 2N$ para la interpredicción.

65 En ejemplos donde la unidad de selección de modo no divide además una CU en PU, cada CU puede asociarse con un bloque de codificación de luma y bloques de codificación de croma correspondientes. Como se indicó anteriormente, el tamaño de una CU puede referirse al tamaño del bloque de codificación de luma de la CU. El codificador de vídeo 200 y el decodificador de vídeo 120 pueden admitir tamaños de CU de $2N \times 2N$, $2N \times N$ o $N \times 2N$.

Para otras técnicas de codificación de vídeo tales como una codificación de modo de copia intrabloque, una codificación de modo afin y una codificación de modo de modelo lineal (LM), como pocos ejemplos, la unidad de selección de modo 202, a través de las unidades respectivas asociadas con las técnicas de codificación, genera un bloque de predicción para el bloque actual que se codifica. En algunos ejemplos, tal como la codificación de modo de paleta, la unidad de selección de modo 202 puede no generar un bloque de predicción, y en su lugar generar elementos de sintaxis que indican la manera en que reconstruir el bloque en base a una paleta seleccionada. En tales modos, la unidad de selección de modo 202 puede proporcionar estos elementos de sintaxis a la unidad de codificación por entropía 220 para que se codifiquen.

Como se describió anteriormente, la unidad de generación residual 204 recibe los datos de vídeo para el bloque actual y el bloque de predicción correspondiente. La unidad de generación residual 204 genera entonces un bloque residual para el bloque actual. Para generar el bloque residual, la unidad de generación residual 204 calcula las diferencias muestra a muestra entre el bloque de predicción y el bloque actual.

La unidad de procesamiento de transformación 206 aplica una o más transformaciones al bloque residual para generar un bloque de coeficientes de transformación (denominado en la presente descripción "bloque de coeficientes de transformación"). La unidad de procesamiento de transformación 206 puede aplicar varias transformaciones a un bloque residual para formar el bloque de coeficientes de transformación. Por ejemplo, la unidad de procesamiento de transformación 206 puede aplicar una transformada de coseno discreta (DCT), una transformada direccional, una transformada de Karhunen-Loeve (KLT) o una transformada conceptualmente similar a un bloque residual. En algunos ejemplos, la unidad de procesamiento de transformación 206 puede realizar múltiples transformaciones a un bloque residual, por ejemplo, una transformada primaria y una transformada secundaria, tal como una transformada rotacional. En algunos ejemplos, la unidad de procesamiento de transformación 206 no aplica transformaciones a un bloque residual.

La unidad de cuantificación 208 puede cuantificar los coeficientes de transformación en un bloque de coeficientes de transformación, para producir un bloque de coeficientes de transformación cuantificado. La unidad de cuantificación 208 puede cuantificar los coeficientes de transformación de un bloque de coeficientes de transformación de acuerdo con un valor de parámetro de cuantificación (QP) asociado con el bloque actual. El codificador de vídeo 200 (por ejemplo, a través de la unidad de selección de modo 202) puede ajustar el grado de cuantificación aplicado a los bloques de coeficiente de transformación asociados con el bloque actual mediante el ajuste del valor de QP asociado con la CU. La cuantificación puede introducir pérdida de información y, por lo tanto, los coeficientes de transformación cuantificados pueden tener una precisión menor que los coeficientes de transformación originales producidos por la unidad de procesamiento de transformación 206.

La unidad de cuantificación inversa 210 y la unidad de procesamiento de transformada inversa 212 pueden aplicar cuantificación inversa y transformadas inversas a un bloque de coeficiente de transformada cuantificado, respectivamente, para reconstruir un bloque residual a partir del bloque de coeficiente de transformada. La unidad de reconstrucción 214 puede producir un bloque reconstruido correspondiente al bloque actual (aunque potencialmente con algún grado de distorsión) en base al bloque residual reconstruido y a un bloque de predicción generado por la unidad de selección de modo 202. Por ejemplo, la unidad de reconstrucción 214 puede añadir muestras del bloque residual reconstruido a muestras correspondientes del bloque de predicción generado por la unidad de selección de modo 202 para producir el bloque reconstruido.

La unidad de filtro 216 puede realizar una o más operaciones de filtro en bloques reconstruidos. Por ejemplo, la unidad de filtro 216 puede realizar operaciones de desbloqueo para reducir los artefactos de bloqueo a lo largo de los bordes de las CU. Las operaciones de la unidad de filtrado 216 pueden omitirse, en algunos ejemplos.

El codificador de vídeo 200 almacena bloques reconstruidos en DPB 218. Por ejemplo, en ejemplos donde no se necesitan las operaciones de la unidad de filtro 216, la unidad de reconstrucción 214 puede almacenar bloques reconstruidos en el DPB 218. En ejemplos donde se necesitan las operaciones de la unidad de filtrado 216, la unidad de filtrado 216 puede almacenar los bloques reconstruidos filtrados en el DPB 218. La unidad de estimación de movimiento 222 y la unidad de compensación de movimiento 224 pueden recuperar una imagen de referencia del DPB 218, formada a partir de los bloques reconstruidos (y potencialmente filtrados), para interpredecir bloques de imágenes codificadas subsecuentemente. Además, la unidad de intrapredicción 226 puede usar bloques reconstruidos en el DPB 218 de una imagen actual para intraprededir otros bloques en la imagen actual.

En general, la unidad de codificación por entropía 220 puede codificar por entropía elementos de sintaxis recibidos de otros componentes funcionales del codificador de vídeo 200. Por ejemplo, la unidad de codificación de entropía 220 puede codificar por entropía bloques de coeficientes de transformación cuantificados de la unidad de cuantificación 208. Como otro ejemplo, la unidad de codificación de entropía 220 puede codificar por entropía elementos de sintaxis de predicción (por ejemplo, información de movimiento para la interpredicción o información de modo interno para la intrapredicción) de la unidad de selección de modo 202. La unidad de codificación por entropía 220 puede realizar una o más operaciones de codificación por entropía en los elementos de sintaxis, que son otro ejemplo de datos de vídeo, para generar datos codificados por entropía. Por ejemplo, la unidad de codificación por entropía 220 puede realizar

una operación de codificación de longitud de variable adaptativa al contexto (CAVLC), una operación CABAC, una operación de codificación de longitud de variable a variable (V2V), una codificación aritmética binaria adaptativa al contexto basada en sintaxis (SBAC), una operación de codificación por Entropía de Particionamiento de Rango de Probabilidad (PIPE), una operación de codificación Exponential-Golomb, u otro tipo de operación de codificación por entropía en los datos. En algunos ejemplos, la unidad de codificación de entropía 220 puede funcionar en modo derivación donde los elementos de sintaxis no se codifican por entropía.

El codificador de vídeo 200 puede emitir un flujo de bits que incluye los elementos de sintaxis codificados por entropía necesarios para reconstruir bloques de un segmento o imagen. En particular, la unidad de codificación de entropía 220 puede emitir el flujo de bits.

Las operaciones descritas anteriormente se describen con respecto a un bloque. Tal descripción debe entenderse como operaciones para un bloque de codificación de luma y/o bloques de codificación de croma. Como se describió anteriormente, en algunos ejemplos, el bloque de codificación de luma y los bloques de codificación de croma son componentes de luma y croma de una CU. En algunos ejemplos, el bloque de codificación de luma y los bloques de codificación de croma son componentes de luma y croma de una PU.

En algunos ejemplos, las operaciones realizadas con respecto a un bloque de codificación de luma no necesitan repetirse para los bloques de codificación de croma. Como ejemplo, las operaciones para identificar un vector de movimiento (MV) y una imagen de referencia para un bloque de codificación de luma no necesitan repetirse para identificar un MV y una imagen de referencia para los bloques de croma. Más bien, el MV para el bloque de codificación de luma puede escalarse para determinar el MV para los bloques de croma, y la imagen de referencia puede ser la misma. Como otro ejemplo, el proceso de intrapredicción puede ser el mismo para el bloque de codificación de luma y los bloques de codificación de croma.

El codificador de vídeo 200 representa un ejemplo de un dispositivo configurado para codificar datos de vídeo que incluyen una memoria configurada para almacenar datos de vídeo, y una o más unidades de procesamiento implementadas en circuitos y configuradas para almacenar información de movimiento para una primera línea de unidad de árbol de codificación (CTU) en una primera memoria intermedia de predictor de vectores de movimiento (MVP) de la memoria, y almacenar información de movimiento para una segunda línea de CTU en una segunda memoria intermedia de MVP histórico de la memoria, la segunda línea de CTU es diferente de la primera línea de CTU.

El codificador de vídeo 200 también representa un ejemplo de un dispositivo configurado para codificar datos de vídeo que incluyen una memoria configurada para almacenar datos de vídeo, y una o más unidades de procesamiento implementadas en circuitos y configuradas para almacenar información de movimiento codificada en un historial de un predictor de vectores de movimiento (MVP), almacenar un tipo diferente de información de movimiento en la memoria intermedia de MVP histórico, y codificar la información de movimiento de un bloque de datos de vídeo mediante el uso de la información de movimiento de la memoria intermedia de MVP histórico.

El codificador de vídeo 200 también representa un ejemplo de un dispositivo configurado para codificar datos de vídeo que incluye una memoria configurada para almacenar datos de vídeo, y una o más unidades de procesamiento implementadas en circuitos y configuradas para almacenar una pluralidad de tipos diferentes de información de movimiento en las respectivas, diferentes memorias intermedias de predicción de vectores de movimiento (MVP).

La Figura 15 es un diagrama de bloques que ilustra un decodificador de vídeo 300 ilustrativo que puede realizar las técnicas de esta descripción. La Figura 15 se proporciona con fines de explicación y no es limitante de las técnicas como se ejemplifica y describe ampliamente en esta descripción. Para fines de explicación, esta descripción describe el decodificador de vídeo 300 de acuerdo con las técnicas de JEM y HEVC. Sin embargo, las técnicas de esta descripción pueden realizarse mediante dispositivos de codificación de vídeo que se configuran de acuerdo con otros estándares de codificación de vídeo.

En el ejemplo de la Figura 15, el decodificador de vídeo 300 incluye la memoria de la memoria intermedia de imágenes codificadas (CPB) 320, la unidad de decodificación de entropía 302, la unidad de procesamiento de predicción 304, la unidad de cuantificación inversa 306, la unidad de procesamiento de transformada inversa 308, la unidad de reconstrucción 310, la unidad de filtro 312 y la memoria intermedia de imágenes decodificadas (DPB) 314. Cualquiera o todas las memorias CPB 320, la unidad de decodificación por entropía 302, la unidad de procesamiento de predicción 304, la unidad de cuantificación inversa 306, la unidad de procesamiento de transformada inversa 308, la unidad de reconstrucción 310, la unidad de filtro 312 y el DPB 314 pueden implementarse en uno o más procesadores o en circuitos de procesamiento. Además, el decodificador de vídeo 300 puede incluir procesadores o circuitos de procesamiento adicionales o alternativos para realizar estas y otras funciones.

La unidad de procesamiento de predicción 304 incluye la unidad de compensación de movimiento 316 y la unidad de intrapredicción 318. La unidad de procesamiento de predicción 304 puede incluir unidades de adición para realizar la predicción de acuerdo con otros modos de predicción. Como ejemplos, la unidad de procesamiento de predicción 304 puede incluir una unidad de paleta, una unidad de copia intrabloque (que puede formar parte de la unidad de

compensación de movimiento 316), una unidad afin, una unidad de modelo lineal (LM) o similar. En otros ejemplos, el decodificador de vídeo 300 puede incluir más, menos, o diferentes componentes funcionales.

La memoria CPB 320 puede almacenar datos de vídeo, tales como un flujo de bits de vídeo codificado, para ser decodificados por los componentes del decodificador de vídeo 300. Los datos de vídeo almacenados en la memoria CPB 320 pueden obtenerse, por ejemplo, del medio legible por ordenador 110 (Figura1). La memoria CPB 320 puede incluir un CPB que almacena datos de vídeo codificados (por ejemplo, elementos de sintaxis) de un flujo de bits de vídeo codificado. Además, la memoria CPB 320 puede almacenar datos de vídeo distintos de los elementos de sintaxis de una imagen codificada, tales como datos temporales que representan las salidas de las diversas unidades del decodificador de vídeo 300. DPB 314 generalmente almacena imágenes decodificadas, que el decodificador de vídeo 300 puede emitir y/o usar como datos de vídeo de referencia cuando decodifica datos o imágenes subsecuentes del flujo de bits de vídeo codificado. La memoria CPB 320 y el DPB 314 pueden formarse por cualquiera de una variedad de dispositivos de memoria, tales como la memoria de acceso aleatorio dinámico (DRAM), que incluye la DRAM síncrona (SDRAM), la RAM magnetorresistiva (MRAM), la RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. La memoria CPB 320 y el DPB 314 pueden proporcionarse por el mismo dispositivo de memoria o por dispositivos de memoria separados. En varios ejemplos, la memoria CPB 320 puede estar integrada en el chip con otros componentes del decodificador de vídeo 300, o fuera del chip con relación a esos componentes.

Adicional o alternativamente, en algunos ejemplos, el decodificador de vídeo 300 puede recuperar datos de vídeo codificados de la memoria 120 (Figura1). Es decir, la memoria 120 puede almacenar datos como se analizó anteriormente con la memoria CPB 320. Igualmente, la memoria 120 puede almacenar instrucciones que se ejecutarán por el decodificador de vídeo 300, cuando parte o toda la funcionalidad del decodificador de vídeo 300 se implementa en software para ser ejecutado por el circuito de procesamiento del decodificador de vídeo 300.

Las diversas unidades mostradas en la Figura 15 se ilustran para ayudar a comprender las operaciones realizadas por el decodificador de vídeo 300. Las unidades pueden implementarse como circuitos de función fija, circuitos programables, o una de sus combinaciones. De manera similar a la Figura 14, los circuitos de función fija se refieren a circuitos que proporcionan una funcionalidad particular y se preestablecen en las operaciones que pueden realizarse. Los circuitos programables se refieren a circuitos que pueden programarse para realizar diversas tareas y proporcionar funcionalidad flexible en las operaciones que pueden realizarse. Por ejemplo, los circuitos programables pueden ejecutar software o microprograma que hacen que los circuitos programables funcionen de la manera definida por las instrucciones del software o microprograma. Los circuitos de función fija pueden ejecutar instrucciones de software (por ejemplo, para recibir parámetros o parámetros de salida), pero los tipos de operaciones que realizan los circuitos de función fija son generalmente inmutables. En algunos ejemplos, una o más de las unidades pueden ser bloques de circuito distintos (de función fija o programable), y en algunos ejemplos, una o más unidades pueden ser circuitos integrados.

El decodificador de vídeo 300 puede incluir ALU, EFU, circuitos digitales, circuitos analógicos y/o núcleos programables formados a partir de circuitos programables. En ejemplos donde las operaciones del decodificador de vídeo 300 se realizan mediante software que se ejecuta en los circuitos programables, la memoria en chip o fuera de chip puede almacenar instrucciones (por ejemplo, código objeto) del software que recibe y ejecuta el decodificador de vídeo 300.

La unidad de decodificación por entropía 302 puede recibir datos de vídeo codificados del CPB y decodificar por entropía los datos de vídeo para reproducir elementos de sintaxis. La unidad de procesamiento de predicción 304, la unidad de cuantificación inversa 306, la unidad de procesamiento de transformación inversa 308, la unidad de reconstrucción 310 y la unidad de filtro 312 pueden generar los datos de vídeo decodificados en base a los elementos de sintaxis extraídos del flujo de bits.

En general, el decodificador de vídeo 300 reconstruye una imagen sobre una base de bloque por bloque. El decodificador de vídeo 300 puede realizar una operación de reconstrucción en cada bloque individualmente (donde el bloque que se está reconstruyendo actualmente, es decir, decodificado, puede denominarse "bloque actual").

La unidad de decodificación por entropía 302 puede decodificar por entropía elementos de sintaxis que definen coeficientes de transformación cuantificados de un bloque de coeficientes de transformación cuantificado, así como también información de transformación, tal como un parámetro de cuantificación (QP) y/o indicación(es) de modo de transformación. La unidad de cuantificación inversa 306 puede usar el QP asociado con el bloque de coeficiente de transformación cuantificado para determinar un grado de cuantificación y, igualmente, un grado de cuantificación inversa para que la unidad de cuantificación inversa 306 lo aplique. La unidad de cuantificación inversa 306 puede, por ejemplo, realizar una operación de desplazamiento a la izquierda a nivel de bits para cuantificar inversamente los coeficientes de transformación cuantificados. La unidad de cuantificación inversa 306 puede de esta manera formar un bloque de coeficientes de transformación que incluye coeficientes de transformación.

Después de que la unidad de cuantificación inversa 306 forma el bloque de coeficientes de transformación, la unidad de procesamiento de transformación inversa 308 puede aplicar una o más transformaciones inversas al bloque de coeficientes de transformación para generar un bloque residual asociado con el bloque actual. Por ejemplo, la unidad

de procesamiento de transformación inversa 308 puede aplicar una DCT inversa, una transformación entera inversa, una transformación Karhunen-Loeve inversa (KLT), una transformación rotacional inversa, una transformación direccional inversa, u otra transformación inversa al bloque de coeficientes.

5 Además, la unidad de procesamiento de predicción 304 genera un bloque de predicción de acuerdo con los elementos de sintaxis de la información de predicción que fueron decodificados por la unidad de decodificación de entropía 302. Por ejemplo, si los elementos de sintaxis de la información de predicción indican que el bloque actual se interpredice, la unidad de compensación de movimiento 316 puede generar el bloque de predicción. En este caso, los elementos de sintaxis de la información de predicción pueden indicar una imagen de referencia en el DPB 314 desde la cual
10 recuperar un bloque de referencia, así como también un vector de movimiento que identifica una ubicación del bloque de referencia en la imagen de referencia con relación a la ubicación del bloque actual en la imagen actual. La unidad de compensación de movimiento 316 generalmente puede realizar el proceso de interpredicción de una manera que es sustancialmente similar a la descrita con respecto a la unidad de compensación de movimiento 224 (Figura 14).

15 De acuerdo con las técnicas de esta descripción, la memoria intermedia de imágenes decodificadas 314 puede incluir una o más memorias intermedias de MVP histórico para líneas de CTU. Es decir, a cada línea CTU se le puede asignar su propia memoria intermedia de MVP, o se puede usar una única memoria intermedia MVP para una pluralidad de líneas CTU. En cualquier caso, el decodificador de vídeo 300 puede restablecer una memoria intermedia de MVP para una línea CTU al comienzo de la codificación de datos de vídeo de la línea CTU. La unidad de compensación de movimiento 316 u otra unidad del decodificador de vídeo 300 puede configurarse para almacenar solo vectores de movimiento únicos en la memoria intermedia de MVP. Como se analizó anteriormente, la unidad de compensación de movimiento 316 u otra unidad del decodificador de vídeo 300 puede configurarse para usar una regla FIFO para
20 gestionar la información de movimiento almacenada en la memoria intermedia de MVP, de manera que cuando se añade un vector de movimiento a la memoria intermedia de MVP, si la memoria intermedia de MVP está llena, la unidad de compensación de movimiento 316 puede eliminar un vector de movimiento añadido más temprano de la memoria intermedia de MVP. En algunos ejemplos, el decodificador de vídeo 300 puede mantener diferentes memorias intermedias de MVP respectivos para cada uno de una variedad de modelos de movimiento, tales como, por ejemplo, modelo de movimiento afín, información de movimiento de modo de copia intrabloque, información de movimiento de compensación de iluminación local, MVP de subbloque y predicción de movimiento temporal.

30 Como otro ejemplo, si los elementos de sintaxis de la información de predicción indican que el bloque actual se predice de forma intra, la unidad de intrapredicción 318 puede generar el bloque de predicción de acuerdo con un modo de intrapredicción indicado por los elementos de sintaxis de la información de predicción. Nuevamente, la unidad de intrapredicción 318 puede realizar generalmente el proceso de intrapredicción de una manera que es sustancialmente similar a la descrita con respecto a la unidad de intrapredicción 226 (Figura 14). La unidad de intrapredicción 318
35 puede recuperar datos de muestras vecinas al bloque actual del DPB 314.

La unidad de reconstrucción 310 puede reconstruir el bloque actual mediante el uso del bloque de predicción y el bloque residual. Por ejemplo, la unidad de reconstrucción 310 puede añadir muestras del bloque residual a las muestras correspondientes del bloque de predicción para reconstruir el bloque actual.

40 La unidad de filtro 312 puede realizar una o más operaciones de filtro en bloques reconstruidos. Por ejemplo, la unidad de filtro 312 puede realizar operaciones de desbloqueo para reducir los artefactos de bloqueo a lo largo de los bordes de los bloques reconstruidos. Las operaciones de la unidad de filtrado 312 no se realizan necesariamente en todos los ejemplos.

45 El decodificador de vídeo 300 puede almacenar los bloques reconstruidos en el DPB 314. Por ejemplo, en ejemplos donde no se necesitan las operaciones de la unidad de filtro 312, la unidad de reconstrucción 310 puede almacenar bloques reconstruidos en el DPB 314. En ejemplos donde se necesitan operaciones de la unidad de filtrado 312, la unidad de filtrado 312 puede almacenar los bloques reconstruidos filtrados en DPB 314. Como se analizó anteriormente, DPB 314 puede proporcionar información de referencia, tal como muestras de una imagen actual para la intrapredicción e imágenes decodificadas previamente para la compensación de movimiento posterior, a la unidad de procesamiento de predicción 304. Además, el decodificador de vídeo 300 puede emitir imágenes decodificadas del DPB 314 para su presentación posterior en un dispositivo de visualización, tal como el dispositivo de visualización 118 de la Figura 1.

55 El decodificador de vídeo 300 representa un ejemplo de un dispositivo configurado para decodificar datos de vídeo que incluyen una memoria configurada para almacenar datos de vídeo, y una o más unidades de procesamiento implementadas en circuitos y configuradas para almacenar información de movimiento para una primera línea de unidad de árbol de codificación (CTU) (que puede procesarse por un primer hilo de un proceso de codificación de vídeo) en un primer historial de un predictor de vectores de movimiento (MVP) de la memoria, y almacenar información de movimiento para una segunda línea de CTU (que puede procesarse por un segundo hilo del proceso de codificación de vídeo) en un segundo historial de un MVP de la memoria, la segunda línea de CTU es diferente de la primera línea de CTU. El segundo hilo puede ser diferente del primer hilo.

65 El decodificador de vídeo 300 también representa un ejemplo de un dispositivo configurado para decodificar datos de vídeo que incluyen una memoria configurada para almacenar datos de vídeo, y una o más unidades de procesamiento

implementadas en circuitos y configuradas para almacenar información de movimiento codificada en una memoria intermedia del predictor de vector de movimiento, MVP, histórico en una memoria intermedia, almacenar un tipo diferente de información de movimiento en el historial de una memoria intermedia de MVP, y codificar la información de movimiento de un bloque de datos de vídeo mediante el uso de la información de movimiento del historial de una memoria intermedia de MVP.

El decodificador de vídeo 300 también representa un ejemplo de un dispositivo configurado para decodificar datos de vídeo que incluyen una memoria configurada para almacenar datos de vídeo, y una o más unidades de procesamiento implementadas en circuitos y configuradas para almacenar una pluralidad de tipos diferentes de información de movimiento en respectivos, diferentes memorias intermedias de predicción de vectores de movimiento (MVP) de la historia.

La Figura 16 es un diagrama de flujo que ilustra un procedimiento ilustrativo para codificar un bloque actual de acuerdo con las técnicas de esta descripción. El bloque actual puede comprender una CU de corriente. Aunque se describe con respecto al codificador de vídeo 200 (Figuras 1 y 14), debe entenderse que otros dispositivos pueden configurarse para realizar un procedimiento similar al de la Figura 16.

En este ejemplo, el codificador de vídeo 200 predice inicialmente el bloque actual mediante el uso de información de movimiento (350). Por ejemplo, el codificador de vídeo 200 puede formar un bloque de predicción para el bloque actual mediante el uso de la información de movimiento. El codificador de vídeo 200 puede entonces calcular un bloque residual para el bloque actual (352). Para calcular el bloque residual, el codificador de vídeo 200 puede calcular una diferencia entre el bloque original no codificado y el bloque de predicción para el bloque actual. El codificador de vídeo 200 puede entonces transformar y cuantificar coeficientes del bloque residual (354). A continuación, el codificador de vídeo 200 puede escanear los coeficientes de transformación cuantificados del bloque residual (356). Durante el escaneo, o después del escaneo, el codificador de vídeo 200 puede codificar por entropía los coeficientes y la información de movimiento (358) mediante el uso de las técnicas de esta descripción. El codificador de vídeo 200 puede codificar los coeficientes mediante el uso de CAVLC o CABAC.

El codificador de vídeo 200 puede construir una lista de candidatos de información de movimiento que incluye, por ejemplo, candidatos HMVP, de acuerdo con cualquiera o todas las técnicas de esta descripción, seleccionar un índice candidato que represente un predictor para la información de movimiento del bloque, y codificar por entropía el índice candidato. De acuerdo con las técnicas de esta descripción, el codificador de vídeo 200 restablece una memoria intermedia de MVP antes de usar la memoria intermedia de MVP para almacenar la información de movimiento de una línea CTU correspondiente. En algunos ejemplos, cada línea CTU puede tener su propia memoria intermedia de MVP, o una memoria intermedia de MVP puede usarse para múltiples líneas CTU. Además, el codificador de vídeo 200 puede almacenar múltiples tipos de información de movimiento en las memorias intermedias de MVP, por ejemplo, la misma memoria intermedia o diferentes memorias intermedias de modelo de movimiento respectivos. El codificador de vídeo 200 puede codificar la información de movimiento de un bloque actual mediante el uso de predictores de vectores de movimiento seleccionados de los datos de la memoria intermedia de MVP. El codificador de vídeo 200 puede entonces emitir los datos codificados por entropía del bloque (360), por ejemplo, que incluyen los datos para los coeficientes y la información de movimiento, tal como el índice candidato.

De esta manera, el procedimiento de la Figura 16 representa un ejemplo de un procedimiento que incluye almacenar información de movimiento para una primera unidad de árbol de codificación (CTU) línea de una imagen en una primera memoria intermedia de predictor de vector de movimiento (MVP) de la memoria; restablecer una segunda memoria intermedia de MVP histórico de la memoria; y después de restablecer la segunda memoria intermedia de MVP histórico, almacenar información de movimiento para una segunda línea CTU de la imagen en la segunda memoria intermedia de MVP histórico, la segunda línea CTU que es diferente de la primera línea CTU.

El procedimiento de la Figura 16 también representa un ejemplo de un procedimiento que incluye almacenar información de movimiento en una memoria intermedia del predictor de vector de movimiento, MVP, histórico; almacenar un tipo diferente de información de movimiento en la memoria intermedia de MVP histórico; y codificar la información de movimiento de un bloque de datos de vídeo mediante el uso de la información de movimiento de la memoria intermedia de MVP histórico.

El procedimiento de la Figura 16 también representa un ejemplo de un procedimiento que incluye almacenar una pluralidad de tipos diferentes de información de movimiento en los respectivos y diferentes memorias intermedias de predicción de vectores de movimiento (MVP) de la historia.

La Figura 17 es un diagrama de flujo que ilustra un procedimiento ilustrativo para decodificar un bloque actual de datos de vídeo de acuerdo con las técnicas de esta descripción. El bloque actual puede comprender una CU de corriente. Aunque se describe con respecto al decodificador de vídeo 300 (Figuras 1 y 15), debe entenderse que otros dispositivos pueden configurarse para realizar un procedimiento similar al de la Figura 17.

El decodificador de vídeo 300 puede recibir datos codificados por entropía para el bloque actual, tal como información de predicción codificada por entropía y datos codificados por entropía para coeficientes de un bloque residual

correspondiente al bloque actual (370). Como se analizó anteriormente, la información de predicción codificada por entropía puede incluir, por ejemplo, un índice candidato en una lista de candidatos, que puede incluir candidatos HMVP de acuerdo con las técnicas de esta descripción. El decodificador de vídeo 300 puede decodificar por entropía los datos codificados por entropía para determinar la información de predicción para el bloque actual y reproducir los coeficientes del bloque residual (372). El decodificador de vídeo 300 puede predecir el bloque actual (374), por ejemplo, mediante el uso del modo de interpredicción como se indica mediante la información de predicción para el bloque actual, para calcular un bloque de predicción para el bloque actual.

En particular, el decodificador de vídeo 300 puede construir una lista de candidatos que incluye los candidatos HMVP como se analizó anteriormente, y luego determinar un candidato de la lista de candidatos para usar como predictor de vector de movimiento para el bloque actual mediante el uso del índice de candidato decodificado. De acuerdo con las técnicas de esta descripción, el decodificador de vídeo 300 restablece una memoria intermedia de MVP antes de usar la memoria intermedia de MVP para almacenar la información de movimiento de una línea CTU correspondiente. En algunos ejemplos, cada línea CTU puede tener su propia memoria intermedia de MVP, o una memoria intermedia de MVP puede usarse para múltiples líneas CTU. Además, el decodificador de vídeo 300 puede almacenar múltiples tipos de información de movimiento en las memorias intermedias de MVP, por ejemplo, la misma memoria intermedia o diferentes memorias intermedias de modelo de movimiento respectivos. El decodificador de vídeo 300 puede seleccionar mediante el uso de predictores de vectores de movimiento a partir de los datos de la memoria intermedia de MVP.

El decodificador de vídeo 300 puede entonces reconstruir un vector de movimiento para el bloque actual mediante el uso del predictor de vectores de movimiento, después predecir el bloque actual mediante el uso del vector de movimiento para generar un bloque de predicción. El decodificador de vídeo 300 puede entonces escanear inversamente los coeficientes reproducidos (376), para crear un bloque de coeficientes de transformación cuantificados. El decodificador de vídeo 300 puede entonces cuantificar inversamente y transformar inversamente los coeficientes para producir un bloque residual (378). El decodificador de vídeo 300 puede decodificar finalmente el bloque actual combinando el bloque de predicción y el bloque residual (380).

De esta manera, el procedimiento de la Figura 17 representa un ejemplo de un procedimiento que incluye almacenar información de movimiento para una primera unidad de árbol de codificación (CTU) línea de una imagen en una primera memoria intermedia del predictor de vector de movimiento, MVP, histórico; restablecer una segunda memoria intermedia de MVP histórico de la memoria; y después de restablecer la segunda memoria intermedia de MVP histórico, almacenar información de movimiento para una segunda línea CTU de la imagen en la segunda memoria intermedia de MVP histórico, la segunda línea CTU es diferente de la primera línea CTU.

El procedimiento de la Figura 17 también representa un ejemplo de un procedimiento que incluye almacenar información de movimiento en una memoria intermedia del predictor de vector de movimiento, MVP, histórico; almacenar un tipo diferente de información de movimiento en la memoria intermedia de MVP histórico; y codificar la información de movimiento de un bloque de datos de vídeo mediante el uso de la información de movimiento de la memoria intermedia de MVP histórico.

El procedimiento de la Figura 17 también representa un ejemplo de un procedimiento que incluye almacenar una pluralidad de tipos diferentes de información de movimiento en los respectivos y diferentes memorias intermedias de predicción de vectores de movimiento (MVP) de la historia.

La Figura 18 es un diagrama de flujo que ilustra un procedimiento ilustrativo de codificación (codificación o decodificación) de datos de vídeo de acuerdo con las técnicas de esta descripción. Por ejemplo, el procedimiento de la Figura 18 puede realizarse durante la etapa 350 de la Figura 16 o la etapa 374 de la Figura 17. Para propósitos de ejemplo y explicación, el procedimiento de la Figura 18 se explica con respecto al decodificador de vídeo 300, aunque el codificador de vídeo 200 también puede realizar este o un procedimiento similar.

El decodificador de vídeo 300 puede codificar bloques de una línea CTU inicial de una imagen (390), por ejemplo, mediante el uso de intrapredicción o interpredicción. El decodificador de vídeo 300 almacena la información de movimiento de una unidad de árbol de codificación (CTU) de una línea de una imagen en una primera memoria intermedia (392), por ejemplo, de DPB 314. El decodificador de vídeo 300 puede usar la información de movimiento de la primera memoria intermedia para codificar la información de movimiento usada durante la codificación de interpredicción. En algunos ejemplos, un primer hilo de un proceso de codificación de vídeo realizado por el decodificador de vídeo 300 puede codificar la primera línea de CTU.

El decodificador de vídeo 300 también restablece una segunda memoria intermedia (394), por ejemplo, de DPB 314. El segundo amortiguador puede ser el mismo que el primer amortiguador, o un amortiguador diferente. El decodificador de vídeo 300 también puede codificar bloques de una segunda línea CTU (396). El decodificador de vídeo 300 almacena la información de movimiento de la segunda línea de CTU en la segunda memoria intermedia (398). En algunos ejemplos, un segundo hilo del proceso de codificación de vídeo realizado por el decodificador de vídeo 300 puede codificar la segunda línea de CTU, donde el segundo hilo es diferente del primer hilo.

De esta manera, el procedimiento de la Figura 18 representa un ejemplo de un procedimiento que incluye almacenar información de movimiento para una primera unidad de árbol de codificación (CTU) línea de una imagen en un primer historial de memoria de predictor de vector de movimiento (MVP); restablecer un segundo historial de memoria MVP de la memoria; y después de restablecer el segundo historial de memoria MVP, almacenar información de movimiento para una segunda línea CTU de la imagen en el segundo historial de memoria MVP, la segunda línea CTU que es diferente de la primera línea CTU.

Debe reconocerse que, en función del ejemplo, ciertos actos o eventos de cualquiera de las técnicas que se describen en la presente memoria pueden realizarse en una secuencia diferente, pueden adicionarse, fusionarse u omitirse por completo (por ejemplo, no todos los actos o eventos que se describen son necesario para la práctica de las técnicas). Además, en ciertos ejemplos, los actos o eventos pueden realizarse concurrentemente, por ejemplo, a través de procesamiento de multihilos, procesamiento de interrupciones o múltiples procesadores, en lugar de secuencialmente.

En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, microprograma o cualquiera de sus combinaciones. Si se implementan en software, las funciones pueden almacenarse en o transmitirse como una o más instrucciones o código en un medio legible por ordenador y que se ejecutan mediante una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible, tal como los medios de almacenamiento de datos o los medios de comunicación, que incluyen cualquier medio que facilite transferir un programa informático de un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador generalmente pueden corresponder a (1) medios de almacenamiento legibles por ordenador tangibles que no son transitorios o (2) un medio de comunicación tal como una señal u onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que se pueda acceder por uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en esta divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

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

Las instrucciones pueden ejecutarse mediante uno o más procesadores, tal como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados de aplicación específica (ASIC), matrices de puertas programables en campo (FPGA) u otros circuitos lógicos discretos o integrados equivalentes. En consecuencia, los términos "procesador" y "circuito de procesamiento", como se usan en la presente descripción, pueden referirse a cualquiera de las estructuras anteriores o cualquier otra estructura adecuada para la implementación de las técnicas descritas en la presente descripción. Además, en algunos aspectos, la funcionalidad descrita en la presente memoria puede proporcionarse dentro de módulos de hardware y/o software dedicados configurados para la codificación y decodificación, o incorporarse en un códec combinado. Asimismo, las técnicas se podrían implementar completamente en uno o más circuitos o elementos lógicos.

Las técnicas de esta divulgación pueden implementarse en una amplia variedad de dispositivos o aparatos, que incluyen un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En esta divulgación se describen diversos componentes, módulos o unidades para enfatizar los aspectos funcionales de los dispositivos configurados para realizar las técnicas divulgadas, pero no necesariamente requieren la realización por diferentes unidades de hardware. Más bien, como se describió anteriormente, diversas unidades pueden combinarse en una unidad de hardware de códec o proporcionarse por una colección de unidades de hardware interoperativas, que incluyen uno o más procesadores como se describió anteriormente, junto con software y/o microprograma adecuados.

REIVINDICACIONES

1. Un procedimiento de codificación de datos de vídeo mediante el uso de procesamiento paralelo de frente de onda, comprendiendo el procedimiento:

5 almacenar (392) información de movimiento para una primera hilera de la unidad de árbol de codificación, CTU, (152) en una primera memoria intermedia del predictor de vector de movimiento, MVP, histórico de una memoria, en el que la primera memoria intermedia de MVP histórico se asocia con un segmento de una imagen;

10 reiniciar (394) una segunda memoria intermedia de MVP histórico de la memoria para que esté vacío antes de codificar una segunda hilera CTU (152), en el que la segunda hilera CTU es diferente de la primera hilera CTU y la segunda memoria intermedia de MVP histórico se asocia con el segmento de la imagen; y después de restablecer la segunda memoria intermedia de MVP histórico, almacenar (398) información de movimiento para la segunda hilera CTU (152) en la segunda memoria intermedia de MVP histórico.
2. El procedimiento de la reivindicación 1, en el que la primera memoria intermedia de MVP histórico y la segunda memoria intermedia de MVP histórico son memorias intermedias de MVP separadas para cada una de la primera hilera CTU y la segunda hilera CTU, en el que almacenar la información de movimiento para la primera hilera CTU comprende almacenar, por un primer hilo de un proceso de codificación de vídeo, la información de movimiento para la primera hilera CTU, y en el que almacenar la información de movimiento para la segunda hilera CTU comprende almacenar, por un segundo hilo del proceso de codificación de vídeo, la información de movimiento para la segunda hilera CTU, siendo el primer hilo diferente del segundo hilo.
3. El procedimiento de la reivindicación 1,

25 en el que almacenar la información de movimiento para la primera hilera de CTU comprende almacenar solo la información de movimiento única dentro de la primera memoria intermedia de MVP histórico en la primera memoria intermedia de MVP histórico al evitar almacenar la información de movimiento en la primera memoria intermedia de MVP histórico que ya está almacenada en la primera memoria intermedia de MVP histórico, y

30 en el que almacenar la información de movimiento para la segunda hilera de CTU comprende almacenar solo la información de movimiento única dentro de la segunda memoria intermedia de MVP histórico en la segunda memoria intermedia de MVP histórico al evitar almacenar la información de movimiento en la segunda memoria intermedia de MVP histórico que ya está almacenada en la segunda memoria intermedia de MVP histórico.
4. El procedimiento de la reivindicación 1, que comprende, además:

40 cuando la primera memoria intermedia de MVP histórico se llena, eliminar uno o más vectores de movimiento insertados más antiguos de la primera memoria intermedia de MVP histórico de acuerdo con una regla de primero en entrar, primero en salir, FIFO; y

45 cuando la segunda memoria intermedia de MVP histórico se llena, se eliminan uno o más vectores de movimiento insertados más antiguos de la segunda memoria intermedia de MVP histórico de acuerdo con la regla FIFO.
5. El procedimiento de la reivindicación 1,

50 en el que almacenar la información de movimiento para la primera hilera CTU comprende almacenar la información de movimiento para cada uno de una pluralidad de tipos diferentes de modelos de movimiento en las respectivas, diferentes memorias intermedias de MVP histórico de una primera pluralidad de memorias intermedias de MVP histórico que incluyen la primera memoria intermedia de MVP histórico; y

55 en el que almacenar la información de movimiento para la segunda hilera de CTU comprende almacenar la información de movimiento para cada uno de la pluralidad de tipos diferentes de modelos de movimiento en las respectivas, diferentes memorias intermedias de MVP histórico de una segunda pluralidad de memorias intermedias de MVP histórico que incluyen la segunda memoria intermedia de historial MVP;

60 en el que la pluralidad de diferentes tipos de modelos de movimiento en particular comprende uno o más del modelo de movimiento afín, información de movimiento del modo de copia intrabloque, información de movimiento de compensación de iluminación local, predicción de vectores de movimiento, MVP, de subbloque o predicción de movimiento temporal.
6. El procedimiento de la reivindicación 1, que comprende además generar un MVP a partir de dos o más MVP en una o más memorias intermedias de MVP histórico;

65 en el que los dos o más MVP tienen en particular diferentes tipos de información de movimiento que comprenden dos o más de información de movimiento codificada, información de movimiento de modelo de movimiento afín, información de movimiento de modo de copia intrabloque, información de movimiento de compensación de iluminación local, información de movimiento de MVP de subbloque, información de

movimiento de predictor de movimiento temporal, información de vector de movimiento sintético en base a una información de movimiento de MVP, información de movimiento de MVP espacial, o información de movimiento de MVP temporal.

- 5 7. El procedimiento de la reivindicación 1, que comprende, además:
- 10 decodificar (390) las unidades de codificación, CU, de las CTU de la primera hilera de CTU mediante el uso de la primera memoria intermedia de MVP histórico; y
 decodificar (396) las CU de las CTU de la segunda hilera de CTU mediante el uso de la segunda memoria intermedia de MVP histórico.
- 15 8. El procedimiento de la reivindicación 1, que comprende, además:
- codificar (390) las unidades de codificación, CU, de las CTU de la primera hilera de CTU mediante el uso de la primera memoria intermedia de MVP histórico; y
 codificar (396) las CU de las CTU de la segunda hilera de CTU mediante el uso de la segunda memoria intermedia de MVP histórico.
- 20 9. El procedimiento de la reivindicación 1, que comprende, además:
- codificar la primera información de movimiento de las unidades de codificación, CU, de las CTU de la primera hilera de CTU mediante el uso de la primera memoria intermedia de MVP histórico, en el que
 codificar la primera información de movimiento comprende codificar la primera información de movimiento mediante el uso de modo de fusión o afin de AMVP; y
 25 codificar la segunda información de movimiento de las CU de las CTU de la segunda hilera de CTU mediante el uso de la segunda memoria intermedia de MVP, en el que la codificación de la segunda información de movimiento comprende codificar la segunda información de movimiento mediante el uso de modo de fusión o afin de AMVP; o
- 30 que comprende, además:
- añadir uno o más vectores de movimiento al primer uno o más vectores de movimiento de la primera información de movimiento unidireccional en la primera memoria intermedia de MVP histórico para convertir la primera información de movimiento unidireccional en la primera información de movimiento bidireccional,
 35 y
 añadir uno o más vectores de movimiento al segundo uno o más vectores de movimiento de la segunda información de movimiento unidireccional en la segunda memoria intermedia de MVP histórico para convertir la segunda información de movimiento unidireccional a la segunda información de movimiento bidireccional.
- 40 10. El procedimiento de la reivindicación 1,
- en el que almacenar la información de movimiento para la primera hilera CTU comprende almacenar solo vectores de movimiento que difieren de otros vectores de movimiento dentro de la primera memoria intermedia de MVP histórico mediante un umbral a la primera memoria intermedia de MVP histórico, y
 45 en el que almacenar la información de movimiento para la segunda hilera de CTU comprende almacenar solo vectores de movimiento que difieren de otros vectores de movimiento dentro de la segunda memoria intermedia de MVP histórico mediante el umbral a la segunda memoria intermedia de MVP histórico; o
 en el que almacenar la información de movimiento para la primera hilera de CTU comprende almacenar datos que indican si los vectores de movimiento de la información de movimiento para la primera hilera de CTU se asocian con unidades de codificación, CU, codificadas en modo de fusión, o CU codificadas AMVP,
 50 y
 en el que almacenar la información de movimiento para la segunda hilera de CTU comprende almacenar datos que indican si los vectores de movimiento de la información de movimiento para la segunda hilera de CTU se asocian con las CU codificadas en modo de fusión o las CU codificadas AMVP.
- 55 11. Un dispositivo (200; 300) para codificar datos de vídeo mediante el uso de procesamiento paralelo de frente de onda, comprendiendo el dispositivo:
- 60 una memoria (218; 314) configurada para almacenar datos de vídeo; y
 una o más unidades de procesamiento implementadas en circuitos y configuradas para:
- almacenar (392) información de movimiento para una primera hilera de la unidad de árbol de codificación, CTU, (152) en una primera memoria intermedia del predictor de vector de movimiento, MVP, histórico de la memoria, en el que la primera memoria intermedia de MVP histórico está asociada con un segmento de una imagen;
- 65

reiniciar (394) una segunda memoria intermedia de MVP de la memoria para que se vacíe antes de codificar una segunda hilera CTU (152), en el que la segunda hilera CTU es diferente de la primera hilera CTU y la segunda memoria intermedia de MVP histórico se asocia con el segmento de la imagen; y
después de restablecer la segunda memoria intermedia de MVP histórico, almacenar (398) información de movimiento para la segunda hilera de CTU en la segunda memoria intermedia de MVP histórico.

12. El dispositivo de la reivindicación 11, en el que la primera memoria intermedia de MVP histórico y la segunda memoria intermedia de MVP histórico son memorias intermedias de MVP separadas para cada una de la primera hilera CTU y la segunda hilera CTU, en el que el uno o más procesadores ejecutan un primer hilo de un proceso de codificación de vídeo para almacenar la información de movimiento para la primera hilera CTU, y en el que el uno o más procesadores ejecutan un segundo hilo del proceso de codificación de vídeo para almacenar la información de movimiento para la segunda hilera CTU, siendo el primer hilo diferente del segundo hilo.

13. El dispositivo de la reivindicación 11, en el que la una o más unidades de procesamiento se configuran además para realizar el procedimiento de una cualquiera de las reivindicaciones 3 a 10.

14. El dispositivo de la reivindicación 11, que comprende además una pantalla (118) configurada para mostrar los datos de vídeo o una cámara (104) configurada para capturar los datos de vídeo; o

en el que el dispositivo comprende una o más de una cámara, un ordenador, un dispositivo móvil, un dispositivo receptor de difusión, o un decodificador; o
en el que el dispositivo comprende al menos uno de:
un circuito integrado;
un microprocesador; o
un dispositivo de comunicación inalámbrica.

15. Un medio de almacenamiento legible por ordenador que tiene almacenadas en el mismo instrucciones que, cuando se ejecutan, hacen que un procesador realice el procedimiento de una cualquiera de las reivindicaciones 1 a 10.

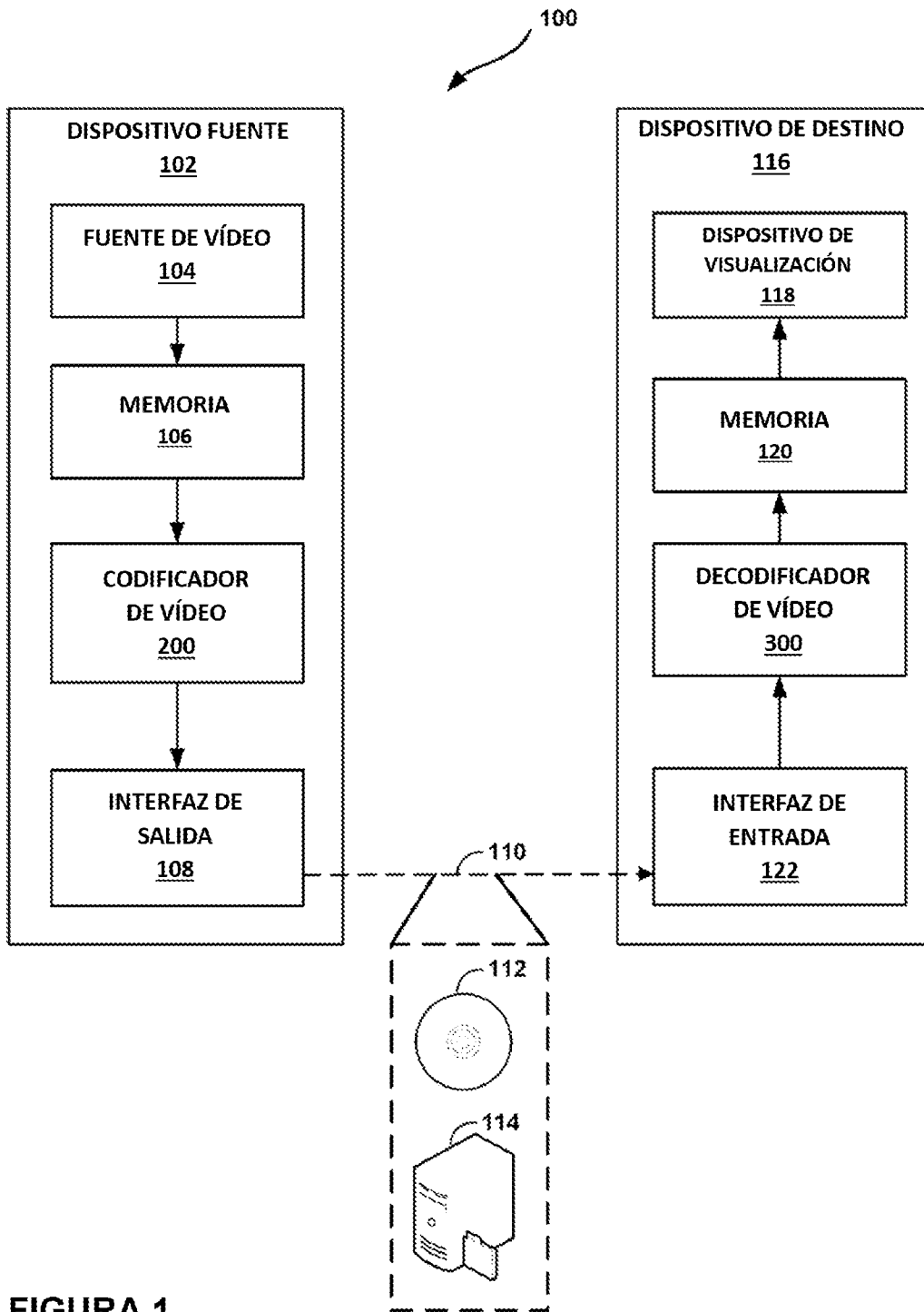


FIGURA 1

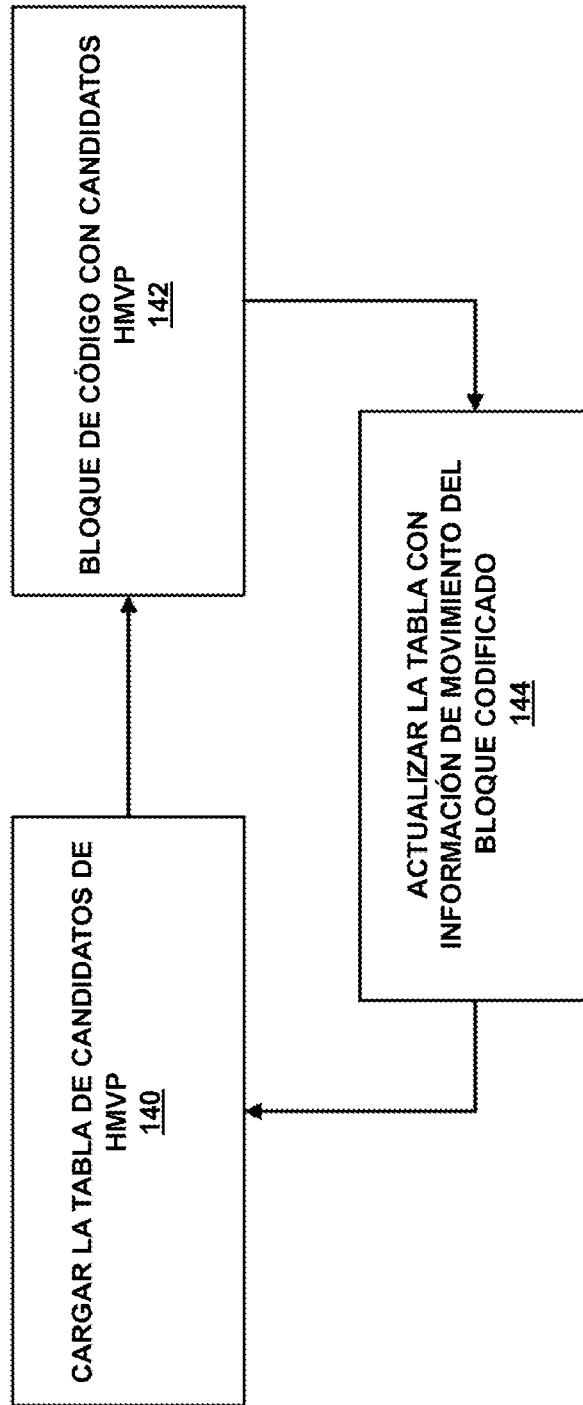


FIGURA 3

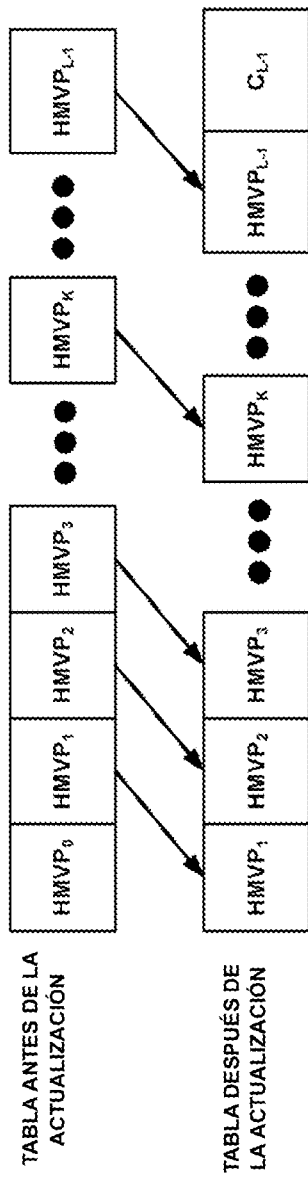


FIGURA 4

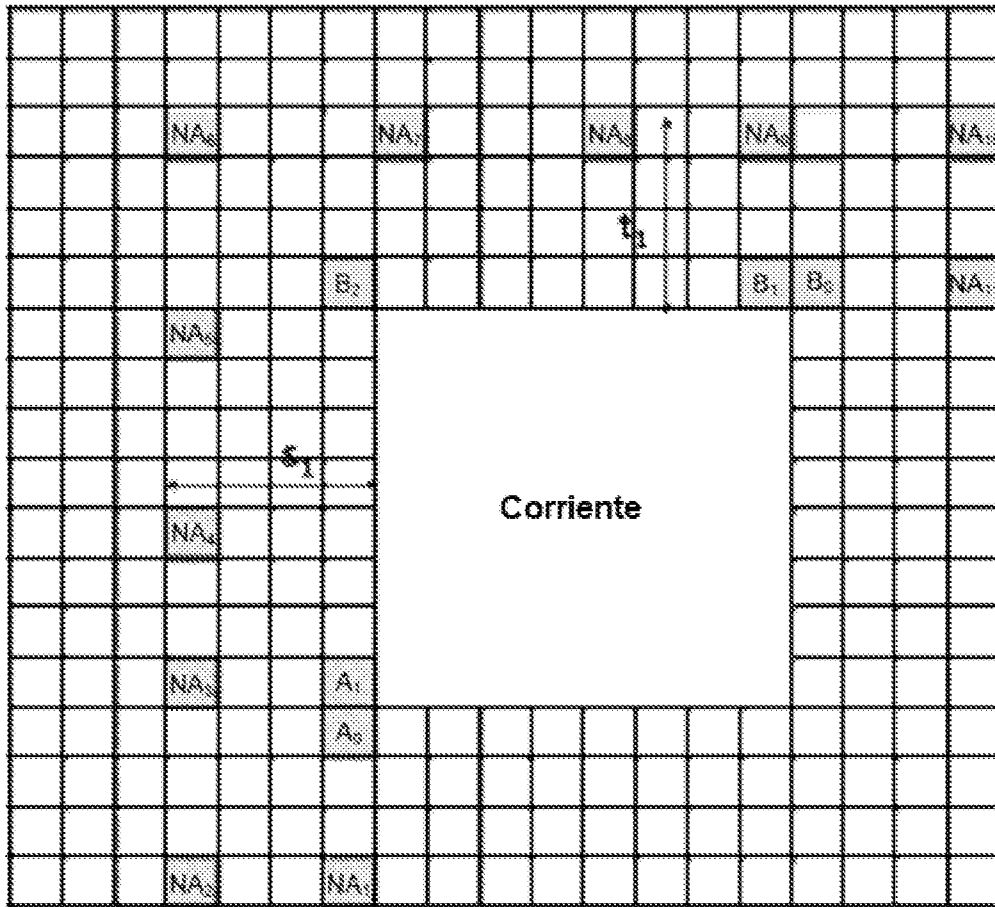


FIGURA 5

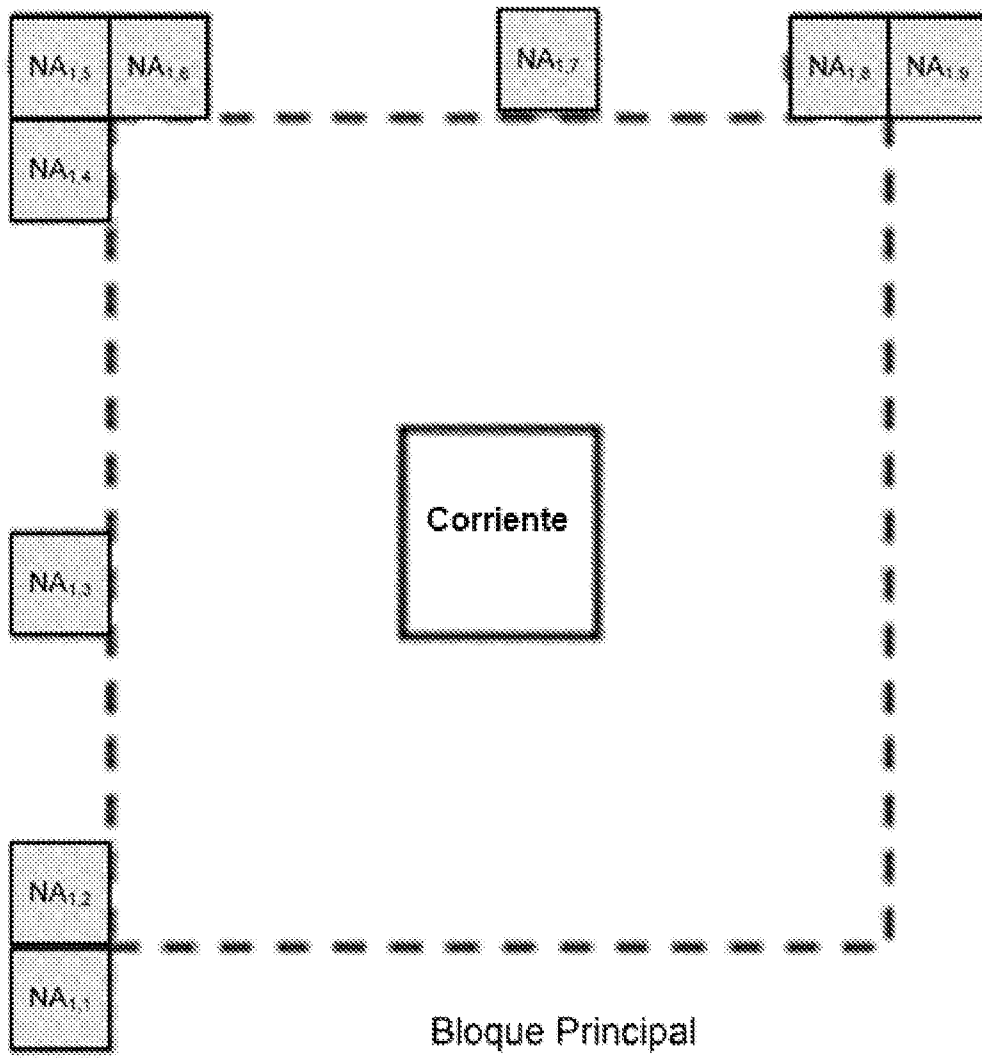


FIGURA 6

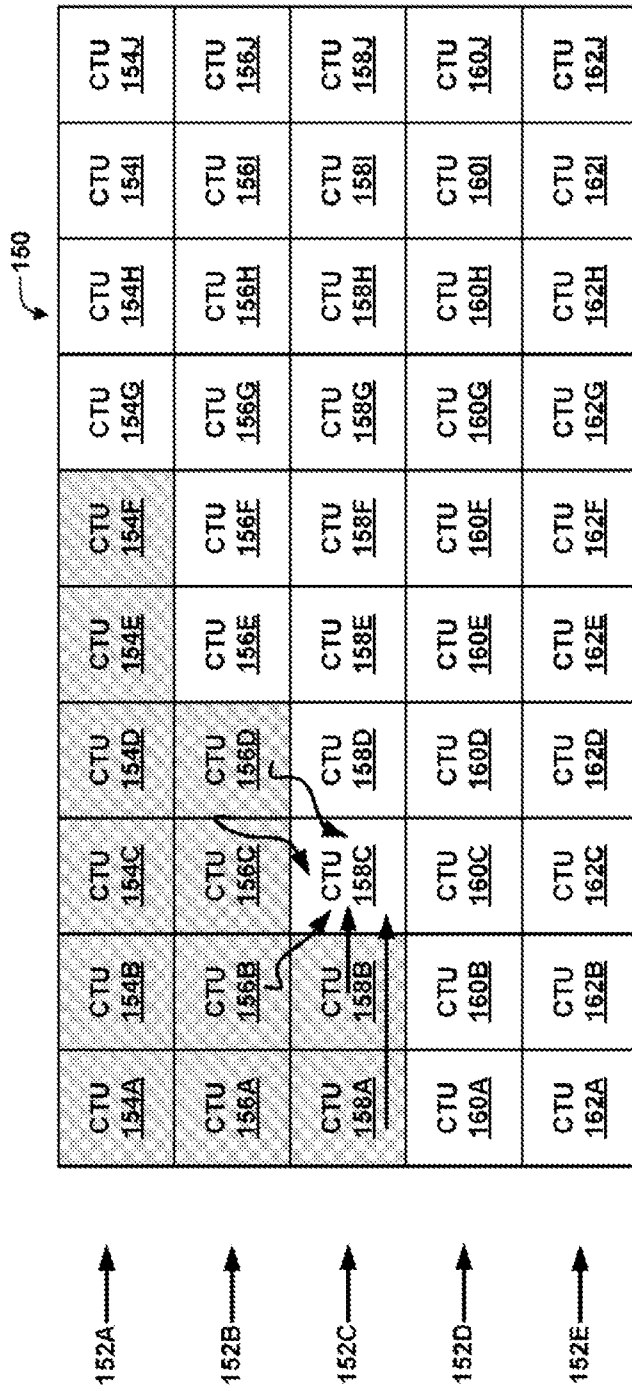


FIGURA 9

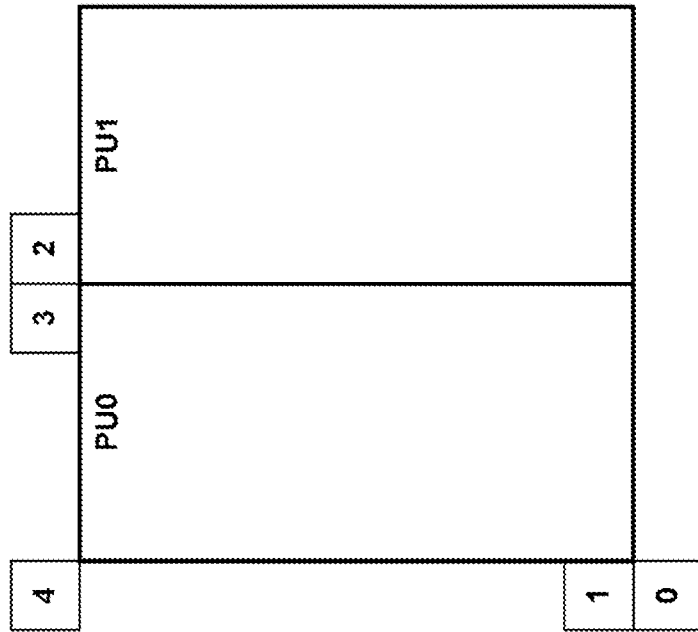


FIGURA 10B

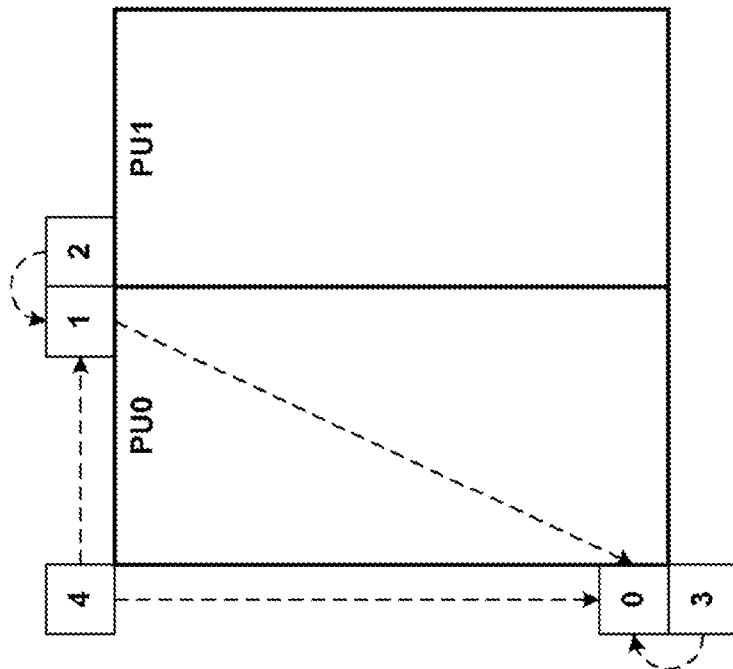


FIGURA 10A

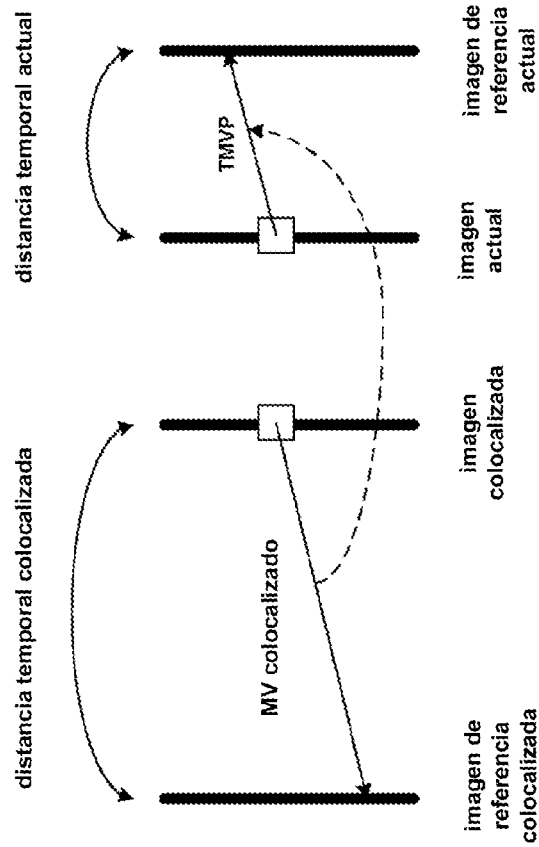


FIGURA 11B

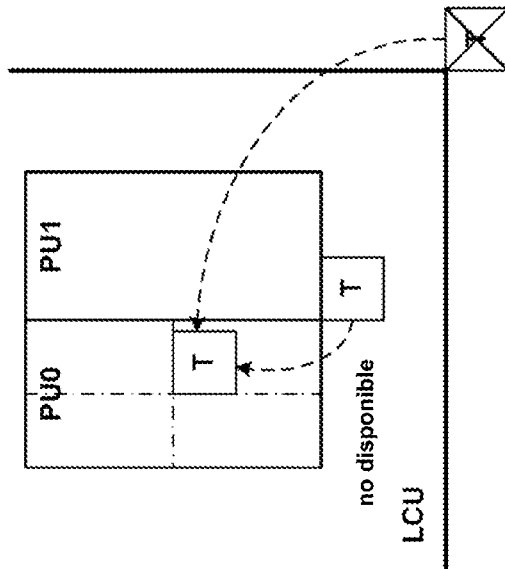


FIGURA 11A

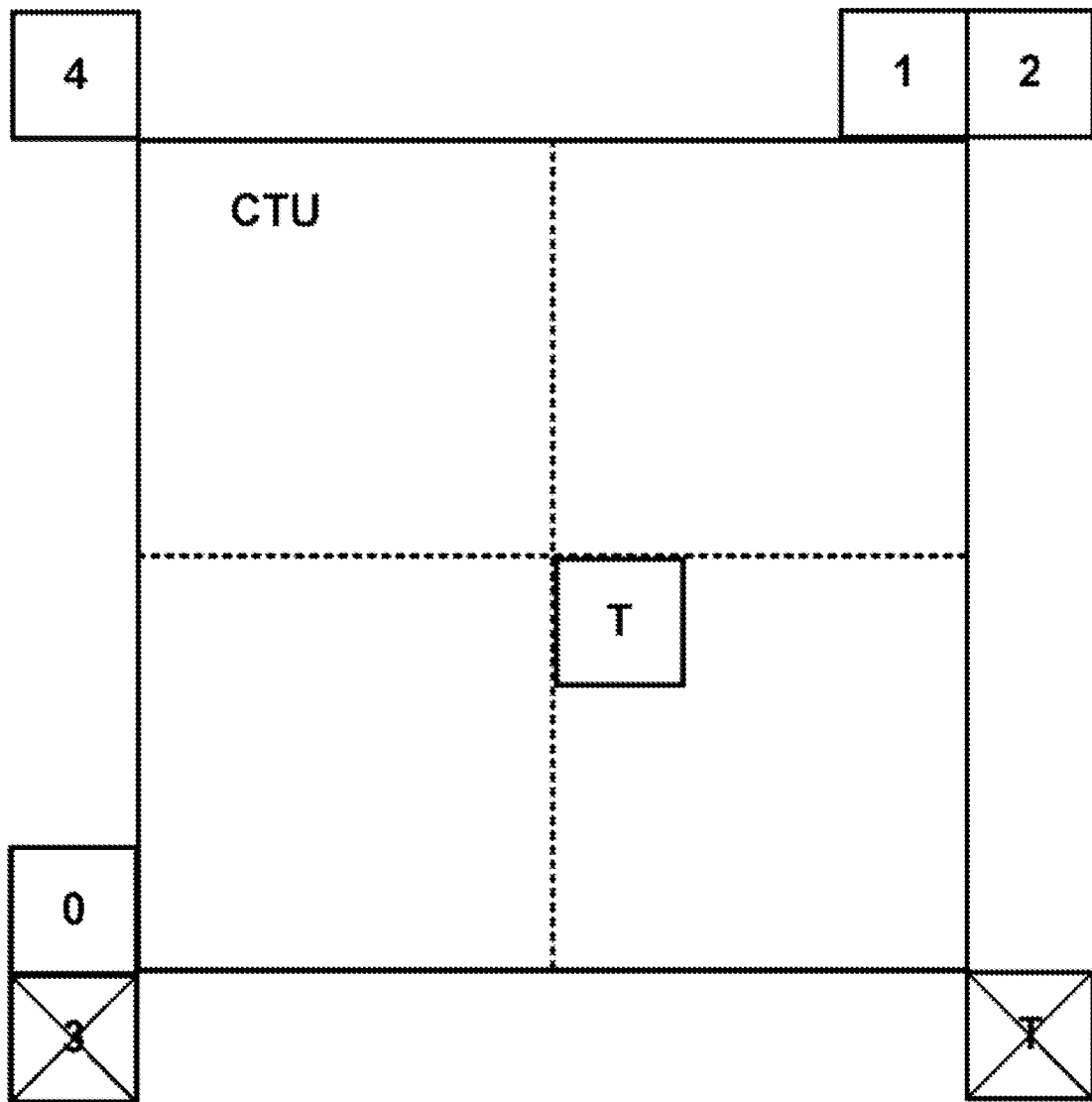


FIGURA 12

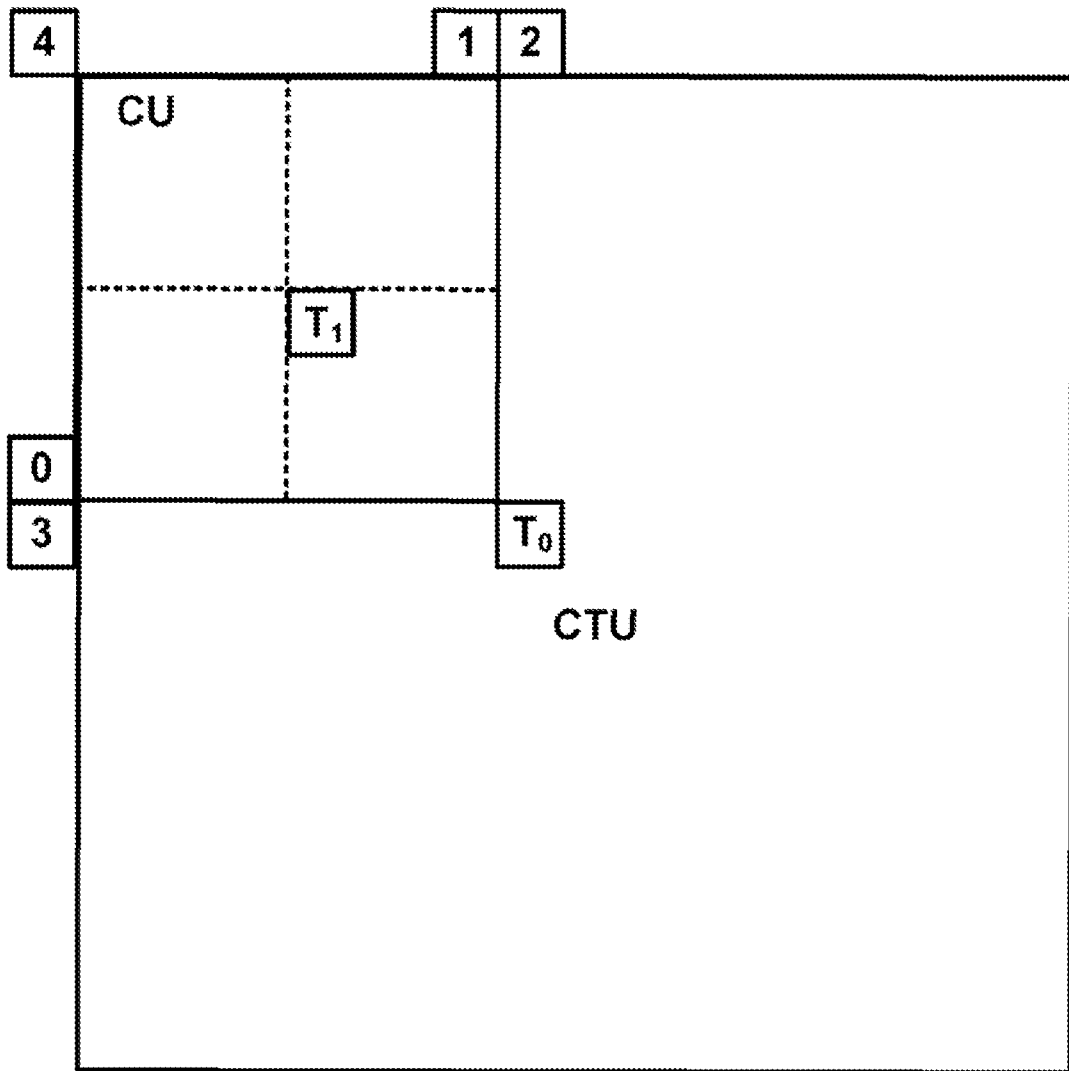


FIGURA 13

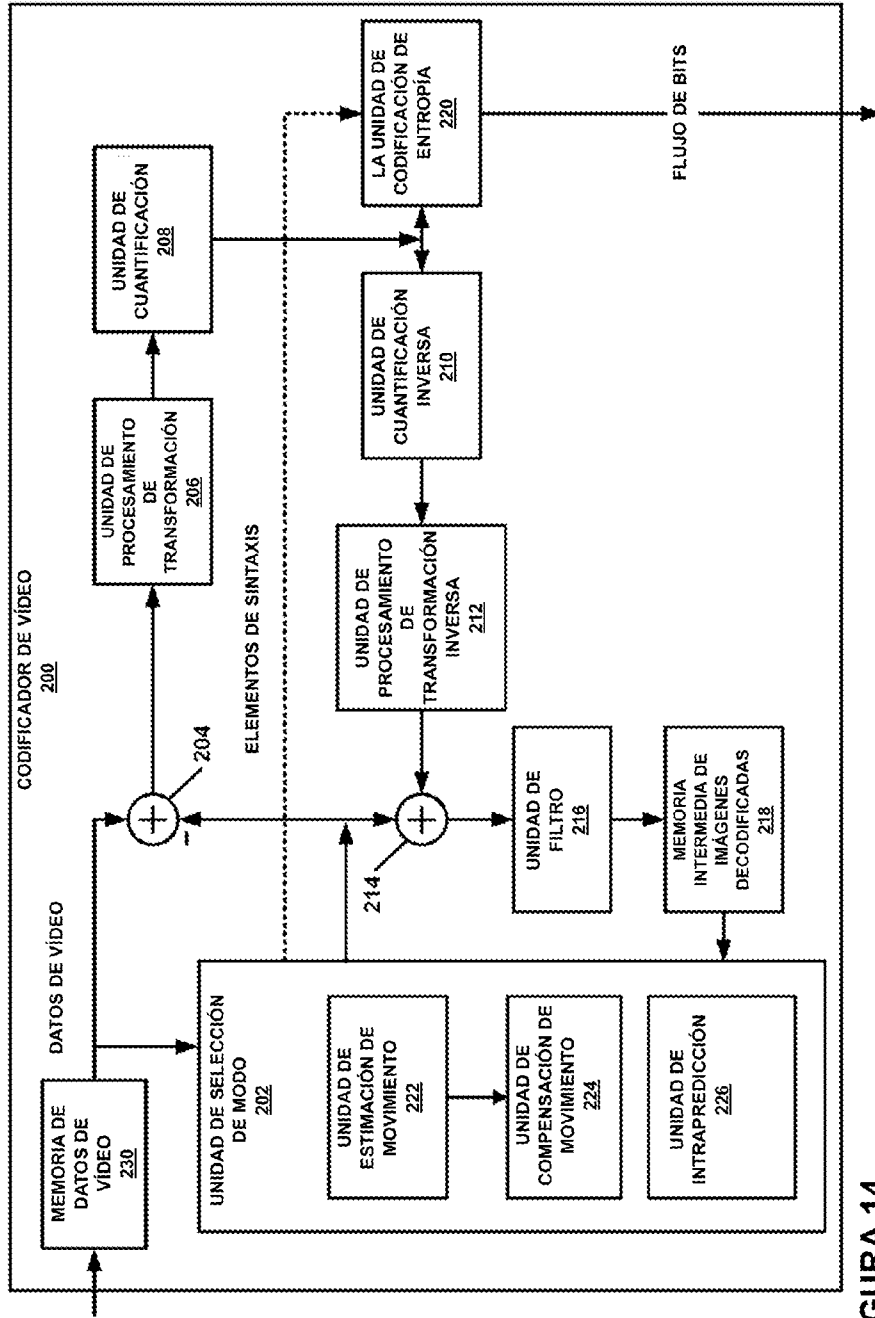


FIGURA 14

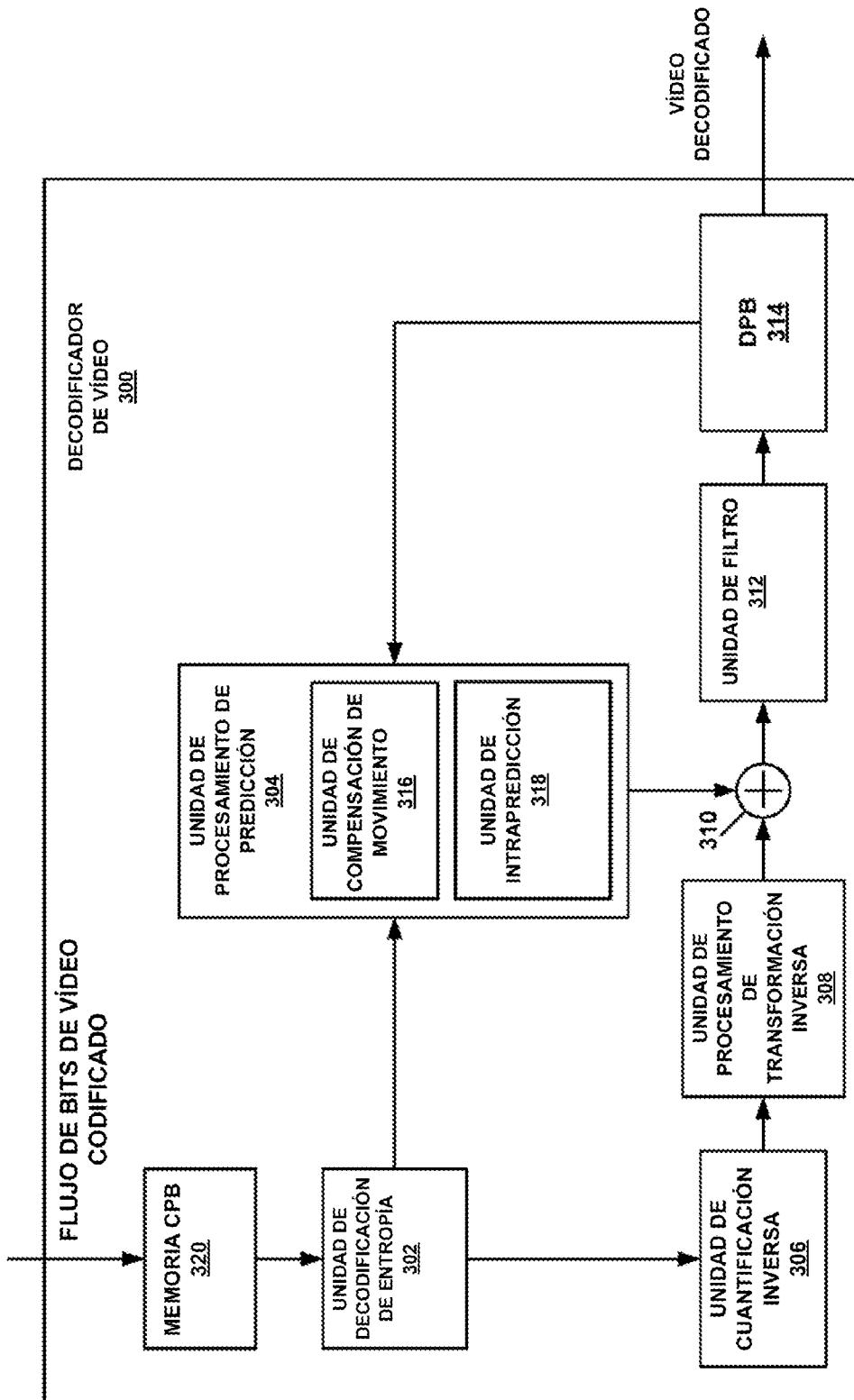


FIGURA 15

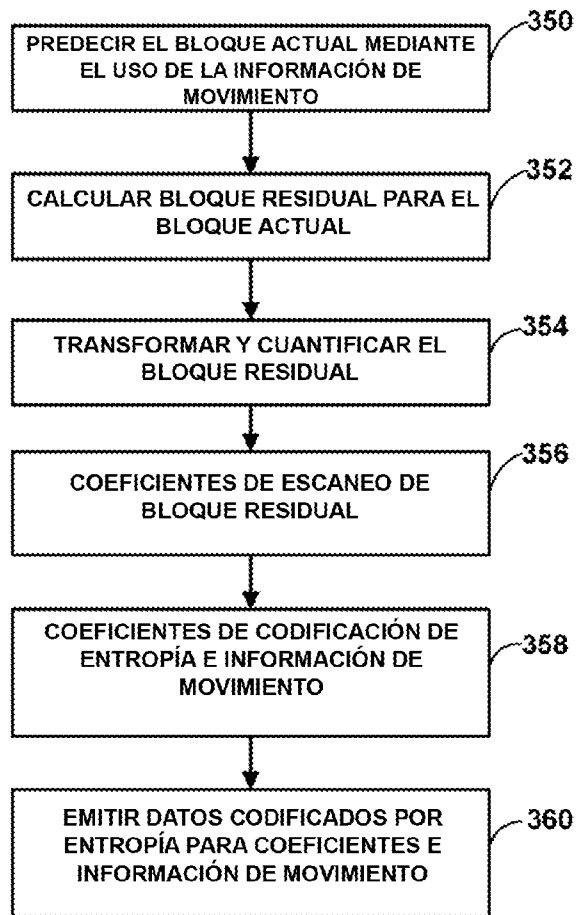


FIGURA 16

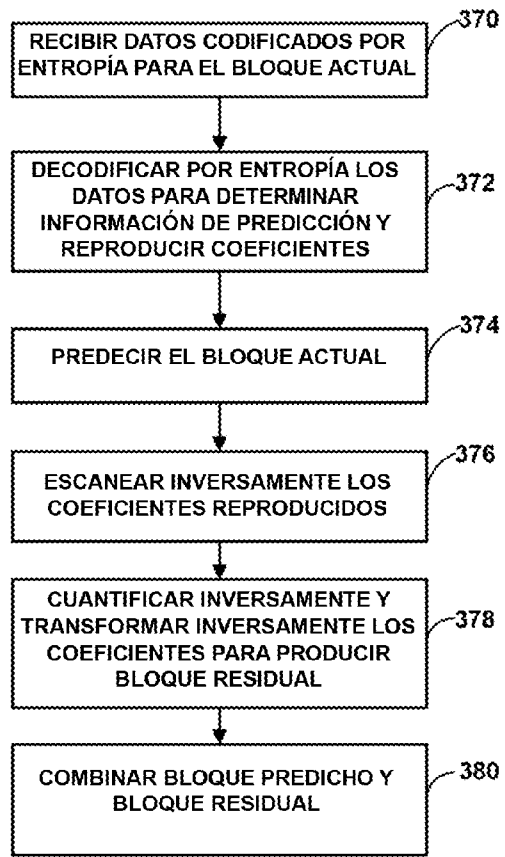


FIGURA 17

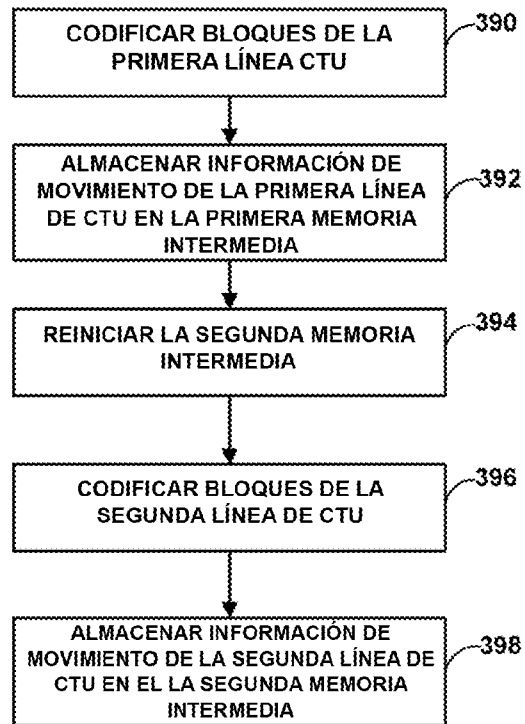


FIGURA 18