

MVP NO ADYACENTES BASADOS EN HISTORIA MÚLTIPLE PARA EL PROCESAMIENTO EN FRENTE DE ONDA DE CODIFICACIÓN DE VIDEO

[0001] La presente solicitud reivindica el beneficio de la Solicitud Provisional US N.º 62/696.281, presentada el 10 de julio de 2018, de la Solicitud Provisional US N.º 62/713.944, presentada el 2 de agosto de 2018 y de la Solicitud US N.º 16/506.720 presentada el 9 de julio de 2019, cuyo contenido completo se incorpora aquí como referencia.

CAMPO TÉCNICO

[0002] La presente divulgación se refiere a la codificación de video, que incluye codificación de video y decodificación de video.

ANTECEDENTES

[0003] Las capacidades de video digital se pueden incorporar en una amplia gama de dispositivos, incluyendo televisores digitales, sistemas de transmisión digital directa, sistemas de transmisión inalámbrica, asistentes digitales personales (PDA), computadoras portátiles o de escritorio, tablets, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos celulares o de radio satelital, los llamados “teléfonos inteligentes”, dispositivos de videoconferencia, dispositivos de transmisión de video, y similares. Los dispositivos de video digital implementan técnicas de codificación de video, 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 video avanzada (AVC), el estándar de codificación de video de alta eficiencia (HEVC), ITU-T H.265/Codificación de video de alta eficiencia (HEVC) y extensiones de dichos estándares. Los dispositivos de video pueden transmitir, recibir, codificar, decodificar y/o almacenar información de video digital de manera más eficiente mediante la implementación de tales técnicas de codificación de video.

[0004] Las técnicas de codificación de video incluyen predicción espacial (intraimagen) y/o predicción temporal (interimagen) para reducir o eliminar la redundancia inherente a las secuencias de video. Para la codificación de video basada en bloques, un segmento de video (por ejemplo, una imagen de video o una porción de una imagen de video) puede dividirse en bloques de video, que también pueden denominarse unidades de árbol de codificación (CTU), unidades de codificación (CU) y/o nodos de codificación. Los bloques de video en un segmento intracodificado (I) de una imagen se codifican

usando predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen. Los bloques de video en una porción de una imagen intercodificada (P o B) pueden usar predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen o predicción temporal con respecto a muestras de referencia en otras imágenes de referencia. Las imágenes pueden denominarse cuadros, y las imágenes de referencia pueden denominarse cuadros de referencia.

SÍNTESIS

[0005] En general, esta divulgación describe técnicas para codificar información de movimiento de bloques de datos de video. Estas técnicas se pueden utilizar durante el procesamiento paralelo de frente de onda. La información de movimiento puede incluir vectores de movimiento que se predicen a partir de predictores de vectores de movimiento históricos (HMVP). Un candidato HMVP puede referirse a la información de movimiento de un bloque previamente codificado. Un codificador de video (codificador o decodificador) puede mantener una tabla con múltiples candidatos HMVP durante un proceso de codificación (codificación o decodificación). El codificador de video puede vaciar la tabla cuando ocurre un nuevo segmento. Cuando hay un bloque intercodificado, el codificador de video puede añadir a la tabla información de movimiento asociada con el bloque intercodificado.

[0006] En un ejemplo, un método de codificación (codificación o decodificación) de datos de video incluye almacenar información de movimiento para una primera línea de unidad de árbol de codificación (CTU) de una imagen en un primer búfer de predicción de vector de movimiento histórico (MVP) de una memoria; restablecer un segundo búfer MVP histórico de la memoria; y después de restablecer el segundo búfer MVP histórico, almacenar información de movimiento para una segunda línea CTU de la imagen en el segundo búfer MVP histórico, siendo la segunda línea CTU diferente de la primera línea CTU. En algunos ejemplos, un primer subproceso de un proceso de codificación de video puede codificar la primera línea CTU y un segundo subproceso del proceso de codificación de video, diferente al primer subproceso, puede codificar la segunda línea CTU.

[0007] En otro ejemplo, un dispositivo para codificar datos de video incluye una memoria configurada para almacenar datos de video; 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) de una imagen en un primer búfer de predicción de vector de movimiento histórico (MVP) de la memoria; restablecer un segundo búfer MVP histórico de la memoria; y después de restablecer el

segundo búfer MVP histórico, almacenar información de movimiento para una segunda línea CTU de la imagen en el segundo búfer MVP histórico, siendo la segunda línea CTU diferente de la primera línea CTU.

[0008] En otro ejemplo, un medio de almacenamiento legible por computadora posee almacenadas en él instrucciones que, cuando se ejecutan, hacen que un procesador: almacene información de movimiento para una primera línea de unidad de árbol de codificación (CTU) de una imagen en un primer búfer predictor de vector de movimiento histórico (MVP) de una memoria; restablecer un segundo búfer MVP histórico de la memoria; y después de restablecer el segundo búfer MVP histórico, almacenar información de movimiento para una segunda línea CTU de la imagen en el segundo búfer MVP histórico, siendo la segunda línea CTU diferente de la primera línea CTU.

[0009] En otro ejemplo, un dispositivo para codificar datos de video incluye una memoria configurada para almacenar datos de video; medios para almacenar información de movimiento para una primera línea de unidad de árbol de codificación (CTU) de una imagen en un primer búfer de predicción de vector de movimiento histórico (MVP) de la memoria; medios para restablecer un segundo búfer MVP histórico de la memoria; y medios para almacenar información de movimiento para una segunda línea CTU de la imagen en el segundo búfer MVP histórico después de restablecer el segundo búfer MVP histórico, siendo la segunda línea CTU diferente de la primera línea CTU.

[0010] Los detalles de uno o más ejemplos se exponen en los dibujos adjuntos y la descripción a continuación. Otras características, objetos y ventajas serán evidentes a partir de la descripción, los dibujos y las reivindicaciones.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

[0011] La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y decodificación de video de ejemplo que puede realizar las técnicas de la presente divulgación.

[0012] Las FIG. 2A y 2B son diagramas conceptuales que ilustran un ejemplo de estructura de árbol binario quadtree (QTBT) y una unidad de árbol de codificación (CTU) correspondiente.

[0013] La FIG. 3 es un diagrama de flujo que ilustra un proceso de ejemplo para codificar información de movimiento utilizando predictores de vector de movimiento histórico (HMVP).

[0014] La FIG. 4 es un diagrama conceptual que ilustra un ejemplo de actualización de una tabla HMVP.

[0015] La FIG. 5 es un diagrama conceptual que ilustra un ejemplo de selección de

bloques no adyacentes para la codificación de información de movimiento.

[0016] La FIG. 6 es un diagrama conceptual que ilustra una selección de ejemplo de bloques no adyacentes basada en un bloque principal.

[0017] La FIG. 7 es un diagrama conceptual que ilustra un ejemplo de un procesamiento de frente de onda deseado de unidades de árbol de codificación (CTU).

[0018] La FIG. 8 es un diagrama conceptual que ilustra un ejemplo de información de movimiento utilizada para HMVP.

[0019] La FIG. 9 es un diagrama conceptual que ilustra un ejemplo de una imagen dividida en múltiples líneas de unidades de árbol de codificación (CTU).

[0020] Las FIGS. 10A y 10B son diagramas de bloques que ilustran ejemplos de vectores de movimiento vecinos espaciales candidatos para los modos de fusión y predicción avanzada de vectores de movimiento (AMVP).

[0021] Las FIGS. 11A y 11B son diagramas conceptuales que ilustran candidatos de predicción de vector de movimiento temporal (TMPV).

[0022] La FIG. 12 es un diagrama de bloques que ilustra un ejemplo de una unidad de árbol de codificación (CTU) y bloques vecinos.

[0023] La FIG. 13 es un diagrama de bloques que ilustra una CU actual en una CTU actual.

[0024] La FIG. 14 es un diagrama de bloques que ilustra un codificador de video de ejemplo que puede realizar las técnicas de esta divulgación.

[0025] La FIG. 15 es un diagrama de bloques que ilustra un decodificador de video de ejemplo que puede realizar las técnicas de esta divulgación.

[0026] La FIG. 16 es un diagrama de flujo que ilustra un método de ejemplo para codificar un bloque actual de datos de video de acuerdo con las técnicas de esta divulgación.

[0027] La FIG. 17 es un diagrama de flujo que ilustra un método de ejemplo para decodificar un bloque actual de datos de video de acuerdo con las técnicas de esta divulgación.

[0028] La FIG. 18 es un diagrama de flujo que ilustra un método de ejemplo de codificación (codificación o decodificación) de datos de video de acuerdo con las técnicas de esta divulgación.

DESCRIPCIÓN DETALLADA

[0029] La FIG. 1 es un diagrama de bloques que ilustra un ejemplo de sistema de codificación y decodificación de video 100 que puede realizar las técnicas de la presente divulgación. Las técnicas de la presente divulgación están generalmente dirigidas a la

codificación (codificación y/o decodificación) de datos de video. En general, los datos de video incluyen cualquier dato para procesar un video. Por lo tanto, los datos de video pueden incluir video sin codificar, video codificado, video decodificado (por ejemplo, reconstruido) y metadatos de video, como datos de señalización.

[0030] Como se muestra en la FIG. 1, el sistema 100 incluye un dispositivo de origen 102 que proporciona datos de video codificados para ser decodificados y visualizados por un dispositivo de destino 116, en este ejemplo. En particular, el dispositivo de origen 102 proporciona los datos de video al dispositivo de destino 116 a través de un medio legible por computadora 110. El dispositivo de origen 102 y el dispositivo de destino 116 pueden comprender cualquiera de una amplia gama de dispositivos, incluyendo computadoras de escritorio, computadoras portátiles (es decir, computadoras portátiles), tablets, decodificadores, microteléfonos como teléfonos inteligentes, televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de transmisión de video o similares. En algunos casos, el dispositivo de origen 102 y el dispositivo de destino 116 pueden estar equipados para la comunicación inalámbrica y, por lo tanto, pueden denominarse dispositivos de comunicación inalámbrica.

[0031] En el ejemplo de la FIG. 1, el dispositivo de origen 102 incluye la fuente de video 104, la memoria 106, el codificador de video 200 y la interfaz de salida 108. El dispositivo de destino 116 incluye la interfaz de entrada 122, el decodificador de video 300, la memoria 120 y el dispositivo de visualización 118. De acuerdo con esta divulgación, el codificador de video 200 del dispositivo de origen 102 y el decodificador de video 300 del dispositivo de destino 116 pueden configurarse para aplicar las técnicas para codificación de información de movimiento. Por lo tanto, el dispositivo de origen 102 representa un ejemplo de un dispositivo de codificación de video, mientras que el dispositivo de destino 116 representa un ejemplo de un dispositivo de decodificación de video. En otros ejemplos, un dispositivo de origen y un dispositivo de destino pueden incluir otros componentes o disposiciones. Por ejemplo, el dispositivo de origen 102 puede recibir datos de video desde una fuente de video externa, tal como una cámara externa. Del mismo modo, el dispositivo de destino 116 puede interactuar con un dispositivo de visualización externo, en lugar de incluir un dispositivo de visualización integrado.

[0032] El sistema 100 como se muestra en la FIG. 1 es simplemente un ejemplo. En general, cualquier dispositivo de codificación y/o decodificación de video digital puede realizar técnicas para codificación de información de movimiento. El dispositivo de

origen 102 y el dispositivo de destino 116 son simplemente ejemplos de tales dispositivos de codificación en los que el dispositivo de origen 102 genera datos de video codificados para su transmisión al dispositivo de destino 116. Esta divulgación 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 video 200 y el decodificador de video 300 representan ejemplos de dispositivos de codificación, en particular, un codificador de video y un decodificador de video, respectivamente. En algunos ejemplos, el dispositivo de origen 102 y el dispositivo de destino 116 pueden funcionar de una manera sustancialmente simétrica de manera que cada uno de los dispositivos de origen 102 y dispositivo de destino 116 incluyen componentes de codificación y decodificación de video. Por lo tanto, el sistema 100 puede soportar la transmisión de video unidireccional o bidireccional entre el dispositivo de origen 102 y el dispositivo de destino 116, por ejemplo, para transmisión de video, reproducción de video, transmisión de video o telefonía de video.

[0033] En general, la fuente de video 104 representa una fuente de datos de video (es decir, datos de video sin codificar) y proporciona una serie secuencial de imágenes (también denominadas “cuadros”) de los datos de video al codificador de video 200, que codifica datos para las imágenes. La fuente de video 104 del dispositivo de origen 102 puede incluir un dispositivo de captura de video, tal como una cámara de video, un archivo de video que contiene video sin procesar previamente capturado, y/o una interfaz de alimentación de video para recibir video de un proveedor de contenido de video. Como alternativa adicional, la fuente de video 104 puede generar datos basados en gráficos de computadora como video fuente, o una combinación de video en vivo, video archivado y video generado por computadora. En cada caso, el codificador de video 200 codifica los datos de video capturados, capturados previamente o generados por computadora. El codificador de video 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 video 200 puede generar un flujo de bits que incluye datos de video codificados. El dispositivo de origen 102 puede entonces emitir los datos de video codificados a través de la interfaz de salida 108 en un medio legible por computadora 110 para recepción y/o recuperación por, por ejemplo, la interfaz de entrada 122 del dispositivo de destino 116.

[0034] La memoria 106 del dispositivo de origen 102 y la memoria 120 del dispositivo de destino 116 representan memorias de uso general. En algunos ejemplos, las memorias 106, 120 pueden almacenar datos de video sin procesar, por ejemplo, video

sin procesar de la fuente de video 104 y datos de video sin decodificar del decodificador de video 300. Adicional o alternativamente, las memorias 106, 120 pueden almacenar instrucciones de *software* ejecutables, por ejemplo, codificador de video 200 y decodificador de video 300, respectivamente. Aunque se muestran por separado del codificador de video 200 y el decodificador de video 300 en este ejemplo, debe entenderse que el codificador de video 200 y el decodificador de video 300 también pueden incluir memorias internas para propósitos funcionalmente similares o equivalentes. Además, las memorias 106, 120 pueden almacenar datos de video codificados, por ejemplo, la salida del codificador de video 200 y la entrada al decodificador de video 300. En algunos ejemplos, partes de las memorias 106, 120 pueden asignarse como uno o más búferes de video, por ejemplo, para almacenar datos de video sin procesar, decodificados y/o codificados.

[0035] El medio legible por computadora 110 puede representar cualquier tipo de medio o dispositivo capaz de transportar los datos de video codificados desde el dispositivo de origen 102 al dispositivo de destino 116. En un ejemplo, el medio legible por computadora 110 representa un medio de comunicación para habilitar el dispositivo de origen 102 para transmitir datos de video codificados directamente al dispositivo de destino 116 en tiempo real, por ejemplo, a través de una red de radiofrecuencia o una red informática. La interfaz de salida 108 puede modular una señal de transmisión que incluye los datos de video 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 inalámbrico o cableado, como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión física. El medio de comunicación puede formar parte de una red basada en paquetes, como una red de área local, una red de área amplia o una red global como Internet. El medio de comunicación puede incluir enrutadores, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo de origen 102 al dispositivo de destino 116.

[0036] En algunos ejemplos, el dispositivo de origen 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, 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 video codificados.

[0037] En algunos ejemplos, el dispositivo de origen 102 puede enviar datos de video codificados al servidor de archivos 114 u otro dispositivo de almacenamiento intermedio que puede almacenar video codificado generado por el dispositivo de origen 102. El dispositivo de destino 116 puede acceder a los datos de video almacenados desde el servidor de archivos 114 mediante transmisión o descarga. El servidor de archivos 114 puede ser cualquier tipo de dispositivo de servidor capaz de almacenar datos de video codificados y transmitir esos datos de video codificados al dispositivo de destino 116. El servidor de archivos 114 puede representar un servidor web (por ejemplo, para un sitio web), un servidor con 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 video codificados desde el servidor de archivos 114 a través de cualquier conexión de datos estándar, incluida una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión Wi-Fi), una conexión por cable (por ejemplo, línea de abonado digital (DSL), módem de cable, etc.) o una combinación de ambos que sea adecuada para acceder a los datos de video codificados almacenados en el servidor de archivos 114. El servidor de archivos 114 y la interfaz de entrada 122 pueden configurarse para funcionar de acuerdo con un protocolo de transmisión de transmisión, un protocolo de transmisión de descarga o una combinación de los mismos.

[0038] 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 los ejemplos en los que 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 se pueden configurar para transferir datos, como datos de video codificados, de acuerdo con un estándar de comunicación celular, como 4G, 4G LTE (Long-Term Evolution), LTE Advanced, 5G o similar. En algunos ejemplos donde la interfaz de salida 108 comprende un transmisor inalámbrico, la interfaz de salida 108 y la interfaz de entrada 122 se pueden configurar para transferir datos, como datos de video codificados, de acuerdo con otros estándares inalámbricos, como una especificación IEEE 802.11, una especificación IEEE 802.15 (por ejemplo, ZigBee™), un estándar Bluetooth™ o similar. En algunos ejemplos, el dispositivo de origen 102 y/o

el dispositivo de destino 116 pueden incluir dispositivos respectivos de sistema en un chip (SoC). Por ejemplo, el dispositivo de origen 102 puede incluir un dispositivo SoC para realizar la funcionalidad atribuida al codificador de video 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 video 300 y/o la interfaz de entrada 122).

[0039] Las técnicas de la presente divulgación se pueden aplicar a la codificación de video en apoyo de cualquiera de una variedad de aplicaciones multimedia, tales como transmisiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de video de transmisión por Internet, tales como transmisión dinámica adaptativa sobre HTTP (DASH), video digital que se codifica en un medio de almacenamiento de datos, decodificación de video digital almacenado en un medio de almacenamiento de datos u otras aplicaciones.

[0040] La interfaz de entrada 122 del dispositivo de destino 116 recibe un flujo de bits de video codificado del medio legible por computadora 110 (por ejemplo, dispositivo de almacenamiento 112, servidor de archivos 114 o similar). El flujo de bits de video codificado puede incluir información de señalización definida por el codificador de video 200, que también es utilizado por el decodificador de video 300, como elementos de sintaxis que tienen valores que describen características y/o el procesamiento de bloques de video 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 video 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 orgánica (OLED) u otro tipo de dispositivo de visualización.

[0041] Aunque no se muestra en la FIG. 1, en algunos ejemplos, el codificador de video 200 y el decodificador de video 300 pueden estar integrados con un codificador de audio y/o decodificador de audio, y pueden incluir unidades MUX-DEMUX apropiadas, u otro *hardware* y/o *software*, para manejar flujos multiplexados que incluyen ambos audio y video en un flujo de datos común. Si corresponde, las unidades MUX-DEMUX pueden cumplir con el protocolo multiplexor ITU H.223 u otros protocolos como el protocolo de datagramas de usuario (UDP).

[0042] El codificador de video 200 y el decodificador de video 300 pueden implementarse como cualquiera de una variedad de circuitos de codificador y/o decodificador adecuados, tales como uno o más microprocesadores, procesadores de

señal digital (DSP), circuitos integrados específicos de aplicación (ASIC), matrices de puerta programables en campo (FPGA), lógica discreta, *software*, *hardware*, *firmware* o cualquier combinación de los mismos. Cuando las técnicas se implementan parcialmente en el *software*, un dispositivo puede almacenar instrucciones para el *software* en un medio adecuado, no transitorio, legible por computadora y ejecutar las instrucciones en *hardware* utilizando uno o más procesadores para realizar las técnicas de la presente divulgación. Cada codificador de video 200 y decodificador de video 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 video 200 y/o un decodificador de video 300 puede comprender un circuito integrado, un microprocesador y/o un dispositivo de comunicación inalámbrico, tal como un teléfono celular.

[0043] El codificador de video 200 y el decodificador de video 300 pueden funcionar de acuerdo con un estándar de codificación de video, tal como ITU-T H.265, también conocido como codificación de video de alta eficiencia (HEVC) o extensiones a los mismos, como la vista múltiple y/o extensiones de codificación de video escalables. Alternativamente, el codificador de video 200 y el decodificador de video 300 pueden operar de acuerdo con otros estándares patentados o industriales, como el modelo de prueba de exploración conjunta (JEM) o ITU-T H.266, también conocido como codificación de video versátil (VVC). Un borrador reciente del estándar VVC se describe en Bross, et al. “Versatile Video Coding (Draft 5),” Joint Video Experts Team (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 (de ahora en más en la presente, “VVC Draft 5”). Sin embargo, las técnicas de esta divulgación no se limitan a ningún estándar de codificación particular.

[0044] En general, el codificador de video 200 y el decodificador de video 300 pueden realizar una codificación de imágenes basada en bloques. El término “bloque” generalmente se refiere a una estructura que incluye datos a procesar (por ejemplo, codificados, decodificados o utilizados 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 video 200 y el decodificador de video 300 pueden codificar datos de video representados en un formato YUV (por ejemplo, Y, Cb, Cr). Es decir, en lugar de codificar datos rojos, verdes y azules (RGB) para muestras de una imagen, el codificador de video 200 y el decodificador de video 300 pueden codificar componentes de luminancia y crominancia,

donde los componentes de crominancia pueden incluir componentes de crominancia de tono rojo y azul. En algunos ejemplos, el codificador de video 200 convierte los datos con formato RGB recibidos en una representación YUV antes de la codificación, y el decodificador de video 300 convierte la representación YUV en el formato RGB. Alternativamente, las unidades de procesamiento previo y posterior (no mostradas) pueden realizar estas conversiones.

[0045] Esta divulgación 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 divulgación 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 video codificado generalmente incluye una serie de valores para elementos de sintaxis representativos de decisiones de codificación (por ejemplo, modos de codificación) y la división de imágenes en bloques. Por lo tanto, las referencias a la codificación de una imagen o un bloque generalmente deben entenderse como valores de codificación para elementos de sintaxis que forman la imagen o el bloque.

[0046] El 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 video (como el codificador de video 200) divide una unidad de árbol de codificación (CTU) en CU según una estructura de árbol cuádruple. Es decir, el codificador de video divide las CTU y CU en cuatro cuadrados iguales y no superpuestos, y cada nodo del quadtree tiene cero o cuatro nodos secundarios. Los nodos sin nodos secundarios pueden denominarse “nodos hoja”, y las CU de dichos nodos hoja pueden incluir una o más PU y/o una o más TU. El codificador de video puede particionar más PU y TU. Por ejemplo, en HEVC, un quadtree 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 predicen de forma interna incluyen información de interpredicción, como una indicación de modo interno.

[0047] Como otro ejemplo, el codificador de video 200 y el decodificador de video 300 pueden configurarse para funcionar de acuerdo con JEM o VVC. Según JEM o VVC, un codificador de video (como el codificador de video 200) divide una imagen en una pluralidad de unidades de árbol de codificación (CTU). El codificador de video 200 puede dividir una CTU de acuerdo con una estructura de árbol, tal como una estructura de árbol binario quadtree (QTBT) o una estructura de árbol de tipo múltiple (MTT). La estructura

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

[0048] En una estructura de partición MTT, los bloques pueden dividirse usando una partición quadtree (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 en la que un bloque se divide en tres sub-bloques. En algunos ejemplos, una partición de árbol triple divide un bloque en tres sub-bloques 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.

[0049] En algunos ejemplos, el codificador de video 200 y el decodificador de video 300 pueden usar una sola estructura QTBT o MTT para representar cada uno de los componentes de luminancia y crominancia, mientras que en otros ejemplos, el codificador de video 200 y el decodificador de video 300 pueden usar dos o más QTBT o MTT, 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 los componentes de crominancia respectivos).

[0050] El codificador de video 200 y el decodificador de video 300 pueden configurarse para usar el particionamiento quadtree por HEVC, particionamiento QTBT de acuerdo con JEM u otras estructuras de particionamiento. Para fines de explicación, la descripción de las técnicas de la presente divulgación se presenta con respecto a la

partición QTBT. Sin embargo, debe entenderse que las técnicas de la presente divulgación también pueden aplicarse a los codificadores de video configurados para usar la partición quadtree u otros tipos de partición.

[0051] Esta divulgación puede usar “NxN” y “N por N” indistintamente para referirse a las dimensiones de muestra de un bloque (como un CU u otro bloque de video) en términos de dimensiones verticales y horizontales, por ejemplo, 16x16 muestras o 16 por 16 muestras. En general, una CU de 16x16 tendrá 16 muestras en dirección vertical ($y = 16$) y 16 muestras en dirección horizontal ($x = 16$). Del mismo modo, una NxN CU generalmente tiene N muestras en dirección vertical y N muestras en dirección horizontal, donde N representa un valor entero no negativo. Las muestras en un CU pueden organizarse en filas y columnas. Además, las CU no necesitan necesariamente tener el mismo número de muestras en la dirección horizontal que en la dirección vertical. Por ejemplo, las CU pueden comprender muestras de NxM, donde M no es necesariamente igual a N.

[0052] El codificador de video 200 codifica datos de video para CU que representan predicción y/o información residual, y otra información. La información de predicción indica cómo se debe predecir la CU para formar un bloque de predicción para la CU. La información residual generalmente representa diferencias muestra por muestra entre muestras de CU antes de la codificación y el bloque de predicción.

[0053] Para predecir una CU, el codificador de video 200 generalmente puede formar un bloque de predicción para la CU a través de la interpredicción o la intrapredicción. La interpredicción generalmente se refiere a la predicción de la CU a partir de datos de una imagen previamente codificada, mientras que la intrapredicción generalmente se refiere a la predicción de la CU a partir de datos previamente codificados de la misma imagen. Para realizar la interpredicción, el codificador de video 200 puede generar el bloque de predicción usando uno o más vectores de movimiento. El codificador de video 200 generalmente puede realizar 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 video 200 puede calcular una métrica de diferencia utilizando una suma de diferencia absoluta (SAD), suma de diferencias cuadradas (SSD), diferencia absoluta media (MAD), diferencias cuadradas medias (MSD) u otros cálculos de diferencia similares para determinar si una referencia el bloque coincide estrechamente con la CU actual. En algunos ejemplos, el codificador de video 200 puede predecir la CU actual usando predicción unidireccional o predicción bidireccional.

[0054] 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 video 200 puede determinar dos o más vectores de movimiento que representan movimiento no traslacional, tal como acercar o alejar, rotación, movimiento en perspectiva u otros tipos de movimiento irregular.

[0055] Para realizar la intrapredicción, el codificador de video 200 puede seleccionar un modo de intrapredicción para generar el bloque de predicción. JEM proporciona sesenta y siete modos de intrapredicción, incluidos varios modos direccionales, así como el modo plano y el modo DC. En general, el codificador de video 200 selecciona un modo de intrapredicción que describe las muestras vecinas a un bloque actual (por ejemplo, un bloque de una CU) desde el cual predecir muestras del bloque actual. Tales muestras generalmente pueden estar arriba, arriba y a la izquierda, o a la izquierda del bloque actual en la misma imagen que el bloque actual, suponiendo que el codificador de video 200 codifique CTU y CU en orden de exploración de trama (de izquierda a derecha, de arriba a abajo).

[0056] El codificador de video 200 codifica datos que representan el modo de predicción para un bloque actual. Por ejemplo, para los modos de interpredicción, el codificador de video 200 puede codificar datos que representan cuál de los diversos modos de predicción disponibles se usa, así como información de movimiento para el modo correspondiente. Para la interpredicción unidireccional o bidireccional, por ejemplo, el codificador de video 200 puede codificar vectores de movimiento utilizando la predicción avanzada de vectores de movimiento (AMVP) o el modo de fusión. El codificador de video 200 puede usar modos similares para codificar vectores de movimiento para el modo de compensación de movimiento afín.

[0057] Después de la predicción, tal como la intrapredicción o la interpredicción de un bloque, el codificador de video 200 puede calcular datos residuales para el bloque. Los datos residuales, como un bloque residual, representan diferencias muestra por muestra entre el bloque y un bloque de predicción para el bloque, formado utilizando el modo de predicción correspondiente. El codificador de video 200 puede aplicar una o más transformadas al bloque residual, para producir datos transformados en un dominio de transformación en lugar del dominio de muestra. Por ejemplo, el codificador de video 200 puede aplicar una transformada discreta de coseno (DCT), una transformada entera, una transformada de ondícula o una transformada conceptualmente similar a los datos de video residual. Además, el codificador de video 200 puede aplicar una transformada secundaria después de la primera transformación, tal como una

transformada secundaria no separable dependiente del modo (MDNSST), una transformada dependiente de la señal, una transformada Karhunen-Loeve (KLT) o similar. El codificador de video 200 produce coeficientes de transformación después de la aplicación de una o más transformadas.

[0058] Como se indicó con anterioridad, después de cualquier transformada para producir coeficientes de transformación, el codificador de video 200 puede realizar la cuantificación de los coeficientes de transformación. La cuantificación generalmente se refiere a un proceso en donde los coeficientes de transformación se cuantifican para posiblemente reducir la cantidad de datos utilizados para representar los coeficientes, proporcionando una compresión adicional. Al realizar el proceso de cuantificación, el codificador de video 200 puede reducir la profundidad de bits asociada con algunos o todos los coeficientes. Por ejemplo, el codificador de video 200 puede redondear un valor de n bits a un valor de m bits durante la cuantificación, donde n es mayor que m . En algunos ejemplos, para realizar la cuantificación, el codificador de video 200 puede realizar un desplazamiento a la derecha del valor a cuantificar.

[0059] Después de la cuantificación, el codificador de video 200 puede escanear los coeficientes de transformación, produciendo un vector unidimensional a partir de la matriz bidimensional que incluye los coeficientes de transformación cuantificados. La exploración puede estar diseñada para colocar coeficientes de transformación de energía más alta (y, por lo tanto, de frecuencia más baja) en la parte delantera del vector y para colocar coeficientes de energía más baja (y, por lo tanto, de frecuencia más alta) en la parte posterior del vector. En algunos ejemplos, el codificador de video 200 puede utilizar un orden de exploración predefinido para explorar los coeficientes de transformación cuantificados para producir un vector serializado, y luego la entropía codifica los coeficientes de transformación cuantificados del vector. En otros ejemplos, el codificador de video 200 puede realizar una exploración adaptativa. Después de escanear los coeficientes de transformación cuantificados para formar el vector unidimensional, el codificador de video 200 puede codificar entropía el vector unidimensional, por ejemplo, de acuerdo con la codificación aritmética binaria adaptativa al contexto (CABAC). El codificador de video 200 también puede codificar valores de entropía para elementos de sintaxis que describen metadatos asociados con los datos de video codificados para uso por el decodificador de video 300 en la decodificación de los datos de video.

[0060] Para realizar CABAC, el codificador de video 200 puede asignar un contexto dentro de un modelo de contexto a un símbolo a transmitir. El contexto puede

relacionarse, por ejemplo, si los valores vecinos del símbolo tienen valor cero o no. La determinación de probabilidad puede basarse en un contexto asignado al símbolo.

[0061] El codificador de video 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, para el decodificador de video 300, por ejemplo, en un encabezado de imagen, un encabezado de bloque, un encabezado de segmento u otros datos de sintaxis, como un conjunto de parámetros de secuencia (SPS), un conjunto de parámetros de imagen (PPS) o un conjunto de parámetros de video (VPS). El decodificador de video 300 también puede decodificar dichos datos de sintaxis para determinar cómo decodificar los datos de video correspondientes.

[0062] De esta manera, el codificador de video 200 puede generar un flujo de bits que incluye datos de video codificados, por ejemplo, elementos de sintaxis que describen la partición de una imagen en bloques (por ejemplo, CU) y predicción y/o información residual para los bloques. Finalmente, el decodificador de video 300 puede recibir el flujo de bits y decodificar los datos de video codificados.

[0063] En general, el decodificador de video 300 realiza un proceso recíproco al realizado por el codificador de video 200 para decodificar los datos de video codificados del flujo de bits. Por ejemplo, el decodificador de video 300 puede decodificar valores para elementos de sintaxis del flujo de bits utilizando CABAC de una manera sustancialmente similar, aunque recíproca, al proceso de codificación de CABAC del codificador de video 200. Los elementos de sintaxis pueden definir información de partición DE una imagen en CTU, y la partición de cada CTU de acuerdo con una estructura de partición correspondiente, como una estructura QTBT, para definir las CU de la CTU. Los elementos de sintaxis pueden definir aún más la predicción y la información residual para bloques (por ejemplo, CU) de datos de video.

[0064] La información residual puede estar representada, por ejemplo, por coeficientes de transformación cuantificados. El decodificador de video 300 puede cuantificar y transformar inversamente los coeficientes de transformación cuantificados de un bloque para reproducir un bloque residual para el bloque. El decodificador de video 300 usa un modo de predicción señalado (intra- o interpredicción) e información de predicción relacionada (por ejemplo, información de movimiento para interpredicción) para formar un bloque de predicción para el bloque. El decodificador de video 300 puede entonces combinar el bloque de predicción y el bloque residual (muestra por muestra) para reproducir el bloque original. El decodificador de video 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.

[0065] Esta divulgación generalmente puede referirse a “señalizar” cierta información, tal como elementos de sintaxis. El término “señalización” puede referirse generalmente a la comunicación de valores DE elementos de sintaxis y/u otros datos utilizados para decodificar datos de video codificados. Es decir, el codificador de video 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 con anterioridad, el dispositivo de origen 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 al almacenar elementos de sintaxis en el dispositivo de almacenamiento 112 para su posterior recuperación por el dispositivo de destino 116.

[0066] De acuerdo con las técnicas de esta divulgación, el codificador de video 200 y el decodificador de video 300 pueden configurarse para realizar un procesamiento paralelo de frente de onda cuando se codifica una imagen de datos de video. En general, el procesamiento paralelo de frente de onda puede implicar la codificación de líneas individuales de unidades de árbol de codificación (CTU) utilizando subprocesos de procesos separados. Por ejemplo, un primer subproceso ejecutado por el codificador de video 200 o el decodificador de video 300 puede procesar una primera línea de CTU, un segundo subproceso puede procesar una segunda línea de CTU, y así sucesivamente. La codificación de una CTU incluye, entre otras cosas, codificar información de movimiento para unidades de codificación de movimiento predicho (CU) de la CTU, que puede referirse a información de movimiento dentro de la misma CTU o una CTU previamente codificada (por ejemplo, una CTU vecina a la izquierda y/o arriba). Tal información de movimiento puede almacenarse en un búfer de predictor de vector de movimiento (MVP). De acuerdo con las técnicas de esta divulgación, el codificador de video 200 y el decodificador de video 300 pueden configurarse para restablecer un búfer MVP para una línea CTU actual antes de codificar datos de video de la línea CTU actual. El búfer MVP puede ser un búfer MVP individual para la línea CTU actual, o puede usarse un búfer MVP común para múltiples líneas de CTU.

[0067] En algunos ejemplos, cuando se almacena información de movimiento en un búfer MVP, el codificador de video 200 y el decodificador 300 de video pueden almacenar solo información de movimiento única dentro del búfer MVP. Por ejemplo, el codificador de video 200 y el decodificador de video 300 pueden codificar una CU actual usando un vector de movimiento actual, determinar si el vector de movimiento está almacenado actualmente en un búfer MVP para la CU actual y, de ser así, evitar

almacenar el vector de movimiento en el búfer MVP, y si no, almacena el vector de movimiento en el búfer MVP.

[0068] En algunos ejemplos, el codificador de video 200 y el decodificador de video 300 pueden usar una regla de primero en entrar, primero en salir (FIFO) para eliminar vectores de movimiento del búfer MVP cuando el búfer MVP se llena. Es decir, para agregar un nuevo vector de movimiento al búfer MVP, el codificador de video 200 y el decodificador de video 300 pueden eliminar un vector de movimiento insertado más temprano del búfer MVP e insertar el nuevo vector de movimiento en el búfer MVP. De esta manera, el búfer MVP puede tener un comportamiento similar a una cola.

[0069] En algunos ejemplos, el codificador de video 200 y el decodificador de video 300 pueden mantener búferes MVP separados para cada una de una variedad de tipos de modelos de movimiento diferentes. Por ejemplo, el codificador de video 200 y el decodificador de video 300 pueden mantener un búfer MVP afín para el modelo de movimiento afín, un búfer MVP de copia intrabloque para información de movimiento del modo de copia intrabloque, un búfer MVP de compensación de iluminación para información de movimiento de compensación de iluminación local, un búfer MVP de sub-bloque para MVP del sub-bloque, y/o un búfer MVP temporal para la predicción del movimiento temporal.

[0070] En algunos ejemplos, el codificador de video 200 y el decodificador de video 300 pueden generar un MVP sintético a partir de dos o más MVP en uno o más búferes de MVP e insertar el MVP sintético en uno de los búferes 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).

[0071] Las FIG. 2A y 2B son diagramas conceptuales que ilustran un ejemplo de estructura 130 de árbol binario quadtree (QTBT), y una unidad de árbol de codificación correspondiente (CTU) 132. Las líneas continuas representan la división de árbol cuadrilátero y las líneas de puntos indican la división de árbol binario. En cada nodo de división (es decir, no hoja) del árbol binario, se señala una bandera para indicar qué tipo de división (es decir, horizontal o vertical) se utiliza, donde 0 indica división horizontal y 1 indica división vertical en este ejemplo. Para la división quadtree, no es necesario indicar el tipo de división, ya que los nodos quadtree dividen un bloque horizontal y verticalmente en 4 sub-bloques con el mismo tamaño. En consecuencia, el codificador de video 200 puede codificar, y el decodificador de video 300 puede decodificar elementos de sintaxis (como la división de información) para un nivel de árbol de región de la estructura QTBT 130 (es decir, las líneas continuas) y elementos de sintaxis (tal

como dividir información) para un nivel de árbol de predicción de la estructura 130 QTBT (es decir, las líneas discontinuas). El codificador de video 200 puede codificar, y el decodificador de video 300 puede decodificar datos de video, tales como predicción y transformación de datos, para CU representadas por nodos hoja terminales de la estructura 130 QTBT.

[0072] En general, CTU 132 de la FIG. 2B puede estar asociado con parámetros que definen tamaños de bloques correspondientes a nodos de la estructura QTBT 130 en el primer y segundo nivel (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 mínimo de quadtree (MinQTSIZE, que representa un tamaño mínimo permitido de nodo de hoja quadtree), un tamaño máximo de árbol binario (MaxBTSIZE, que representa una raíz de árbol binaria máxima permitida tamaño del nodo), una profundidad máxima del árbol binario (MaxBTDepth, que representa una profundidad máxima permitida del árbol binario) y un tamaño mínimo del árbol binario (MinBTSIZE, que representa el tamaño mínimo permitido del nodo de la hoja del árbol binario).

[0073] El nodo raíz de una estructura QTBT correspondiente a una CTU puede tener cuatro nodos secundarios en el primer nivel de la estructura QTBT, cada uno de los cuales puede dividirse de acuerdo con la partición quadtree. Es decir, los nodos del primer nivel son nodos hoja (que no tienen nodos secundarios) o tienen cuatro nodos secundarios. El ejemplo de la estructura 130 QTBT representa nodos tales como el nodo primario y los nodos secundarios que tienen líneas continuas para las ramas. Si los nodos del primer nivel no son mayores que el tamaño máximo permitido del nodo raíz del árbol binario (MaxBTSIZE), entonces los nodos pueden dividirse aún más por los respectivos árboles binarios. La división del árbol binario de un nodo puede repetirse hasta que los nodos resultantes de la división alcancen el tamaño mínimo permitido del nodo de la hoja del árbol binario (MinBTSIZE) o la profundidad máxima permitida del árbol binario (MaxBTDepth). El ejemplo de la estructura 130 QTBT representa los nodos que tienen líneas discontinuas para las ramas. El nodo de hoja de árbol binario se conoce como una unidad de codificación (CU), que se utiliza para la predicción (por ejemplo, predicción intraimagen o interimagen) y transforma, sin ninguna partición adicional. Como se discutió con anterioridad, las CU también pueden denominarse “bloques de video” o “bloques”.

[0074] En un ejemplo de la estructura de partición QTBT, el tamaño de la CTU se establece en 128x128 (muestras de luma y dos muestras de croma 64x64

correspondientes), el MinQTSIZE se establece en 16x16, el MaxBTSIZE se establece en 64x64, el MinBTSIZE (para ambos anchos y altura) se establece en 4, y MaxBTDepth se establece en 4. La partición quadtree se aplica primero a la CTU para generar nodos de hoja de cuatro árboles. Los nodos de hoja quadtree pueden tener un tamaño de 16x16 (es decir, el tamaño MinQTSIZE) a 128x128 (es decir, el tamaño de la CTU). Si el nodo hoja quadtree es 128x128, no se dividirá más por el árbol binario, ya que el tamaño excede el tamaño MaxBTS (es decir, 64x64, en este ejemplo). De lo contrario, el árbol binario dividirá el nodo hoja quadtree. Por lo tanto, el nodo hoja quadtree es también 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 más división. Cuando el nodo del árbol binario tiene un ancho igual a MinBTSIZE (4, en este ejemplo), implica que no se permite más división vertical. De manera similar, un nodo de árbol binario que tiene una altura igual a MinBTSIZE implica que no se permite más división horizontal para ese nodo de árbol binario. Como se señaló con anterioridad, los nodos hoja del árbol binario se denominan CU y se procesan más de acuerdo con la predicción y se transforman sin más particiones.

[0075] La FIG. 3 es un diagrama de flujo que ilustra un proceso de ejemplo para codificar información de movimiento usando predictores de vector de movimiento histórico (HMVP). Inicialmente, un codificador de video, como el codificador de video 200 o el decodificador de video 300, carga una tabla con candidatos HMVP (140). El codificador de video luego codifica un bloque de datos de video usando los candidatos HMVP (142). El codificador de video luego actualiza la tabla con información de movimiento del bloque codificado (144).

[0076] La FIG. 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 entrar, primero en salir (FIFO). La FIG. 4 representa un ejemplo en donde se aplica la regla FIFO para eliminar un candidato HMVP y agregar uno nuevo a la tabla utilizada en los ejemplos de las técnicas de esta divulgación.

[0077] Un codificador de video, tal como un codificador de video 200 o un decodificador de video 300, puede insertar candidatos de HMVP desde la última entrada a 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 video puede aplicar la poda a los candidatos HMVP. El codificador de video 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.

[0078] En el ejemplo de la FIG. 4, la tabla antes de la actualización incluye el historial MVP₀ (HMVP₀) al historial MVP_{L-1} (HMVP_{L-1}), donde los números de subíndice 0 a L-1 representan un orden en donde se agregan los MVP del historial. C_{L-1} representa un nuevo MVP histórico que se agregará a la tabla. Por lo tanto, de acuerdo con la regla FIFO, HMVP₀ se elimina de la tabla antes de agregar C_{L-1}, en este ejemplo.

[0079] La FIG. 5 es un diagrama conceptual que ilustra un ejemplo de selección de bloques no adyacentes para la codificación de información de movimiento. En el ejemplo de la FIG. 5, un bloque actual etiquetado “Curr” representa una unidad de codificación actual (CU), para la cual la información de movimiento puede codificarse usando bloques vecinos adyacentes y/o no adyacentes etiquetados A_i, B_j y NA_k. La predicción de vector de movimiento no adyacente se describe, por ejemplo, en la Solicitud US N.º 16/003.269, presentada el 8 de junio de 2018. Un codificador de video puede aplicar una regla FIFO y un tamaño máximo de búfer de candidatos de movimiento para bloques no adyacentes.

[0080] La FIG. 6 es un diagrama conceptual que ilustra una selección de ejemplo de bloques no adyacentes basada en un bloque principal. Es decir, el bloque principal es un bloque dividido en sub-bloques, incluido el bloque actual. Por ejemplo, el bloque principal puede ser una CTU o un sub-bloque en donde se ha dividido la CTU. Similar a la FIG. 5, en la FIG. 6, una CU actual está etiquetada como “Curr”, y los bloques no adyacentes de los cuales se puede recuperar información de movimiento y usarse para predecir la información de movimiento de la CU actual están etiquetados como “NA_{i,j}”.

[0081] Los vectores de movimiento de bloques espaciales adyacentes de un bloque coubicado pueden usarse como candidatos de predicción de vector 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 la parte inferior derecha del bloque coubicado).

[0082] Las técnicas de esta divulgación pueden utilizarse para mejorar la predicción del vector de movimiento, por ejemplo, agregando candidatos usados para AMVP y/o modos de codificación de fusión, donde los candidatos agregados 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 FIG. 6.

[0083] La FIG. 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 FIG. 7, se pueden asignar varios subprocesos para procesar diferentes líneas de CTU. Es decir, un codificador de video, tal como un codificador de video 200 o un decodificador de video 300, puede ejecutar múltiples subprocesos diferentes para,

por ejemplo, procesamiento paralelo de frente de onda (WPP) al codificar diferentes líneas CTU. En algunos ejemplos, ciertas probabilidades utilizadas para la codificación basada en el contexto (por ejemplo, codificación CABAC), la información de movimiento de los bloques predichos no podría determinarse si las probabilidades se determinaran a partir de un último bloque de una línea CTU anterior, por ejemplo, suponiendo que el último bloque aún no se había codificado. Por tanto, de acuerdo con las técnicas de esta divulgación, el codificador de video 200 y el decodificador de video 300 pueden restablecer un búfer de CTU para una línea de CTU antes de codificar la línea de CTU, para asegurar que las líneas de CTU se puedan procesar correctamente.

[0084] La FIG. 8 es un diagrama conceptual que ilustra un ejemplo de información de movimiento utilizada para HMVP. La FIG. 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 los vectores de movimiento de bloques adicionales podrían considerarse. En particular, en la FIG. 8, X representa la información de movimiento que se está codificando actualmente y los MV de los bloques sombreados están en el búfer histórico. Esta divulgación reconoce que las técnicas HMVP convencionales no utilizan completamente vectores de movimiento de bloques no adyacentes al menos en parte debido al uso de la regla FIFO, como se muestra en la FIG. 8.

[0085] En particular, cuando el bloque X está codificado, la información de movimiento de los bloques no adyacentes (TL0, T0, T1, TR0, TR1, TR2, TR3) de la CTU superior izquierda, la CTU superior y la CTU superior derecha tienen eliminado del búfer del historial. Por lo tanto, la información de movimiento de estos bloques no se considera para agregar 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 el búfer histórico.

[0086] Esta divulgación también reconoce que un único búfer para HVMP no es aplicable para el procesamiento paralelo de frente de onda. Si solo se utiliza un búfer, el tamaño del búfer sería muy alto para contener los posibles candidatos espaciales para los bloques que se procesan en cada subproceso (por ejemplo, la línea CTU). Por ejemplo, si se configuran cuatro subprocesos para que se ejecuten en paralelo, el tamaño del búfer puede llegar a 64. Como resultado, se necesitan más bits para señalar un índice del MVP al decodificador de video 300. Asimismo, pueden ocurrir entradas redundantes. Es decir, una entrada en el búfer histórico puede ser potencialmente útil para un bloque en esta línea, pero podría no ser útil para bloques en otras líneas (por ejemplo, X y F en la Figura 8). En consecuencia, puede resultar difícil encontrar el

candidato óptimo para un bloque.

[0087] La FIG. 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 FIG. 9, la imagen 150 incluye las líneas CTU 152A-152E (líneas CTU 152). Cada una de las líneas 152 de CTU incluye un conjunto respectivo de CTU: la línea 152A de CTU incluye las unidades 154A a 154J, la línea 152B de la CTU incluye las 156A a 156J, la línea 152C de la CTU incluye las 158A a 158J, la línea 152D de la CTU incluye las 160A a 160J y las CTU la línea 152E incluye las CTU 162A – 162J.

[0088] El codificador de video 200 y el decodificador de video 300 pueden configurarse de acuerdo con las técnicas de esta divulgación para utilizar múltiples búferes para MVP basado en historial. En algunos ejemplos, el codificador de video 200 y el decodificador de video 300 pueden mantener búferes MVP histórico separados para cada una de las líneas 152 de CTU (cada una de las cuales puede ser procesada por un subproceso de procesamiento respectivo separado), o puede haber un solo búfer que se restablece en el comienzo de cada línea CTU cuando se aplica el procesamiento paralelo de frente de onda.

[0089] En un ejemplo, CTU 158C puede representar una CTU actual. La información de movimiento de las CTU 154A-154F, 156A-156D, 158A y 158B (mostradas usando sombreado gris en la FIG. 9) puede estar disponible en uno o más búferes MVP históricos respectivos para usar al codificar información de movimiento de CTU 158C.

[0090] Adicional o alternativamente, el codificador de video 200 y el decodificador de video 300 pueden realizar la inicialización de un búfer MVP histórico usando cualquiera o todas las siguientes técnicas, solas o en combinación. El codificador de video 200 y el decodificador de video 300 pueden restablecer el búfer MVP histórico de cada línea CTU para que esté vacía. El codificador de video 200 y el decodificador de video 300 pueden rellenar previamente el búfer MVP histórico de cada línea CTU con vectores de movimiento cero con diferentes índices de cuadro de referencia y/o direcciones de interpredicción, u otra información de movimiento predefinida o derivada. El codificador de video 200 y el decodificador de video 300 pueden rellenar previamente el búfer MVP histórico de cada línea CTU con información de movimiento de los cuadros codificados (imágenes) en la misma capa temporal o capas temporales inferiores (imágenes de referencia que están disponibles para el cuadro/imagen actual).

[0091] El codificador de video 200 y el decodificador de video 300 pueden escalar la información de movimiento, por ejemplo, basándose en 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 video 200 y el decodificador de video 300 pueden combinar la información de movimiento con la información de movimiento del búfer MVP del historial anterior en los cuadros/imágenes codificados, o la información de movimiento de la región coubicada (puede ser CTU o mayor que un determinado tamaño de bloque, por ejemplo, bloque 4x4) en los cuadros/imágenes codificados. El codificador de video 200 y el decodificador de video 300 pueden rellenar previamente el búfer MVP histórico de una línea de CTU anterior cuando se codifica la CTU de arriba a la derecha de la CTU actual. El codificador de video 200 y el decodificador de video 300 pueden usar vectores de movimiento cero con diferentes índices de cuadros de referencia y/o direcciones de predicción entre sí, u otra información de movimiento predefinida o derivada.

[0092] Adicional o alternativamente, el codificador de video 200 y el decodificador de video 300 pueden, siempre que se codifique (codifique o decodifique) una CTU de líneas CTU, usar el búfer MVP histórico asociado para inicializar o modificar los búferes históricos de la línea CTU por debajo de la línea CTU actual.

[0093] Adicional o alternativamente, el codificador de video 200 y el decodificador de video 300 pueden aplicar una regla FIFO para eliminar una entrada de un búfer MVP histórico asociado.

[0094] Adicional o alternativamente, el codificador de video 200 y el decodificador de video 300 pueden borrar un búfer MVP histórico cuando la línea CTU asociada está completamente codificada/decodificada.

[0095] El codificador de video 200 y el decodificador de video 300 pueden mantener un tamaño de búfer de MVP mayor que AMVP/fusión u otras listas de candidatos de modo. Cualquier MV o MV del búfer se pueden seleccionar como candidato(s) de MV para la lista de candidatos utilizada en un modo determinado, por ejemplo, AMPV, modo de fusión, afín o cualquier otro modo inter. Se puede definir una regla sobre cómo seleccionar MV del búfer, por ejemplo, tomar los últimos N MV agregados al búfer, o tomar algunos desde el principio del búfer, y/o algunos desde el medio del búfer, y/o algunos desde el final del búfer. Alternativamente, la señalización se puede aplicar para indicar cuales MV se seleccionan (por ejemplo, el codificador de video 200 puede codificar datos señalizados y el decodificador de video 300 puede decodificar los datos señalizados). El tamaño del búfer de MVP se puede señalar en cualquier conjunto de parámetros (por ejemplo, conjunto de parámetros de video, conjunto de parámetros de secuencia, conjunto de parámetros de imagen o similares), encabezado de segmento o en cualquier otro lugar. El búfer MVP se puede asociar con un segmento, una imagen

y/o una secuencia de video.

[0096] Cuando el codificador de video 200 y el decodificador de video 300 procesan un bloque intercodificado, los MV utilizados en el bloque se pueden añadir al búfer de MVP, y sólo se pueden mantener MV únicos en el búfer. Cuando el búfer está lleno, los MV más antiguos se pueden eliminar del búfer cuando se agrega un nuevo MV. Puede haber una regla por la cual se pueden agregar MV al búfer, por ejemplo, solo se puede agregar MV señalizado, como en el modo AMVP, y si un bloque está codificado en modo de fusión, no se agrega el MV del bloque al búfer.

[0097] El codificador de video 200 y el decodificador de video 300 pueden añadir un MV a MV o MV ya existentes en el búfer. Por ejemplo, si los MV ya existentes en el búfer son unidireccionales, al agregar un nuevo MV, esos MV existentes se pueden modificar para que sean bidireccionales agregando el nuevo MV.

[0098] Se puede aplicar algún procesamiento de MV mientras se agrega un nuevo MV. Por ejemplo, si un nuevo MV está cerca de los MV ya existentes en el búfer, esos MV cercanos se pueden eliminar. Cerrar puede significar cerrar comparando los valores de los componentes de MT (por ejemplo, componentes x e y). En algunos ejemplos, solo los MV que difieren de los MV ya existentes en el búfer por un umbral se pueden agregar al búfer. El mismo umbral se puede configurar para diferentes búferes.

[0099] Los vectores de movimiento en el búfer pueden ser unidireccionales (L0 o L1), bidireccionales o cualquier otro modelo de movimiento MV.

[0100] La información de modo se puede asociar con un MV en el búfer, y si un índice de un MV en el búfer se señala en el bloque o se aplica otra regla sobre la obtención de MV del búfer, entonces la información de modo se puede derivar de 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.

[0101] Esta divulgación reconoce además que los MVP convencionales basados en el historial solo contienen predictores de movimiento regulares y solo se usan para la predicción de movimiento regular sin modificar la información de movimiento. De acuerdo con las técnicas de esta divulgación, el codificador de video 200 y el decodificador de video 300 pueden usar al menos un búfer MVP histórico que contiene no solo información de movimiento codificada sino también otros tipos de predictores de movimiento, como para el modelo de movimiento afín, información de movimiento de modo de copia intrabloque, información de movimiento de compensación de iluminación local, sub-bloque MVP o predictor de movimiento temporal.

[0102] Adicional o alternativamente, el codificador de video 200 y el decodificador de

video 300 pueden usar múltiples búferes MVP histórico para diferentes modelos de movimiento, como modelo de movimiento afín, información de movimiento del modo de copia intrabloque, información de movimiento de compensación de iluminación local, sub-bloque MVP, o predictor de movimiento temporal.

[0103] Adicional o alternativamente, el vector de movimiento sintético basado en el MVP actual y otro predictor de movimiento, como otro MVP espacial o MVP temporal, también se puede añadir a la lista de candidatos.

[0104] Adicional o alternativamente, el codificador de video 200 y el decodificador de video 300 pueden generar un MVP sintético a partir de dos o más MVP en el búfer de MVP histórico o uno o más MVP en el búfer de MVP histórico con otro tipo de MVP, como MVP espaciales o temporales.

[0105] El codificador de video 200 y el decodificador de video 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 muestras, una CTU incluye un bloque $N \times N$ de muestras de luma junto con dos bloques correspondientes de muestras de crominancia. Una CTU se divide en unidades de codificación (CU) utilizando una estructura de árbol. Cada CU de hoja se puede dividir en una, dos o cuatro unidades de predicción (PU) según el tipo de división de PU. Después de obtener el bloque residual aplicando el proceso de predicción basado en el tipo de división de PU, se puede dividir una CU hoja en unidades de transformación (TU).

[0106] En VVC, un árbol cuádruple con árbol de múltiples tipos anidado que usa una estructura de segmentación de divisiones binarias y ternarias reemplaza los conceptos de tipos de unidades de múltiples particiones, es decir, la división de árboles de múltiples tipos anidados 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 de transformación máxima y admiten más flexibilidad para las formas de partición de CU. En la estructura del árbol de codificación, una CU puede tener forma cuadrada o rectangular.

[0107] El codificador de video 200 y el decodificador de video 300 pueden utilizar información de movimiento para predecir un bloque de datos de video. Para cada bloque, puede estar disponible un conjunto de información de movimiento. El conjunto de información de movimiento contiene información de movimiento para 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 “adelante” y “atrás” no necesariamente tienen un significado geométrico. En cambio, se utilizan para distinguir en qué lista de imágenes de referencia se basa un vector de movimiento. Predicción hacia adelante significa la predicción formada sobre la base de la lista de referencia 0, mientras que predicción hacia atrás significa la predicción formada sobre la base de la lista de referencia 1. En los casos en que tanto la lista de referencia 0 como la lista de referencia 1 se utilizan para formar una predicción para un bloque dado, se llama predicción bidireccional.

[0108] Para una imagen o segmento dado, si solo se usa una lista de imágenes de referencia, cada bloque dentro de la imagen o segmento se predice hacia adelante. Si se utilizan ambas listas de imágenes de referencia para una imagen o un segmento determinados, un bloque dentro de la imagen o segmento puede predecirse hacia adelante, predecirse hacia atrás o predecirse bidireccionalmente.

[0109] Para cada dirección de predicción, la información de movimiento contiene un índice de referencia y un vector de movimiento. Se utiliza 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 un componente tanto horizontal como vertical, y cada uno indica un valor de desplazamiento a lo largo de la dirección horizontal y vertical, respectivamente. En algunas descripciones, por simplicidad, la expresión “vector de movimiento” puede usarse indistintamente con información de movimiento, para indicar tanto el vector de movimiento como su índice de referencia asociado.

[0110] El recuento de orden de imágenes (POC) se usa ampliamente en estándares de codificación de video para identificar un orden de visualización de una imagen. Aunque hay casos en los que dos imágenes dentro de una secuencia de video codificada pueden tener el mismo valor de POC, normalmente no sucede dentro de una secuencia de video codificada. Cuando están presentes múltiples secuencias de video codificadas en un flujo de bits, las imágenes con un mismo valor de POC pueden estar más cerca unas de otras en términos de orden de decodificación.

[0111] En HEVC, existen dos modos de interpredicción, denominados modo de fusión (considerándose el salto como un caso especial de modo de fusión) y modo de predicción de vector de movimiento avanzado (AMVP), respectivamente para una PU.

[0112] En el modo AMVP o de fusión, se mantiene una lista de candidatos de vector de movimiento (MV) para múltiples predictores de vector de movimiento. Los vectores de

movimiento, así como los índices de referencia en el modo de fusión, de la PU actual se generan tomando un candidato de la lista de candidatos MV.

[0113] La lista de candidatos de 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 utilizan 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 AMVP, los vectores de movimiento predichos se pueden refinar aún más.

[0114] Como se puede ver anteriormente, 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 se pueden derivar de manera similar a partir de los mismos bloques vecinos espaciales y temporales.

[0115] Las FIGS. 10A y 10B son diagramas de bloques que ilustran ejemplos de vectores de movimiento vecinos espaciales candidatos para los modos de fusión y predicción avanzada de vectores de movimiento (AMVP). La FIG. 10A muestra un ejemplo de candidatos MV espaciales vecinos para el modo de fusión, y la FIG. 10B muestra un ejemplo de candidatos MV espaciales vecinos para el modo AMVP. Los candidatos espaciales MV se derivan de los bloques vecinos como se muestra en las Figs. 10A y 10B. Para una PU específica (PU0), los métodos para generar los candidatos a partir de los bloques difieren para los modos de fusión y AMVP.

[0116] En el modo de fusión, se pueden derivar hasta cuatro candidatos MV espaciales con el orden mostrado en la FIG. 10 A. 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 FIG. 10 A.

[0117] En el modo AVMP, los bloques vecinos se dividen en dos grupos. El primer grupo es un grupo de la izquierda que incluye los bloques 0 y 1. El segundo grupo es un grupo anterior que incluye los bloques 2, 3 y 4, como se muestra en la FIG. 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 máxima prioridad

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 dicho candidato, el primer candidato disponible se escalará para formar el candidato final, por lo que las diferencias de distancia temporal se pueden compensar.

[0118] Las FIGS. 11A y 11B son diagramas conceptuales que ilustran candidatos de predicción de vector de movimiento temporal (TMPV). FIG. 11A muestra un ejemplo de un candidato TMVP. Un candidato de TMVP, si está habilitado y disponible, se agrega a la lista de candidatos de MV después de los candidatos de vector de movimiento espacial. El proceso de derivación del vector de movimiento para un candidato de TMVP es el mismo para los modos de fusión y AMVP, sin embargo, el índice de referencia objetivo para el candidato de TMVP en el modo de fusión siempre se establece en 0.

[0119] La ubicación del bloque principal para la derivación del candidato de TMVP es el bloque inferior derecho fuera de la PU colocada como se muestra en la FIG. 11A como un bloque "T", para compensar el sesgo hacia los bloques de arriba y de la izquierda utilizados para generar candidatos espaciales vecinos. Sin embargo, si ese bloque está ubicado fuera de la fila CTB actual o si la información de movimiento no está disponible, el bloque se sustituye por un bloque central de la PU.

[0120] Un vector de movimiento para un candidato de TMVP se deriva de la PU coubicada de la imagen coubicada, indicada en el nivel de segmento. El vector de movimiento para la PU coubicada se denomina MV coubicado.

[0121] La FIG. 11B muestra un ejemplo de escala de MV. Para derivar el vector de movimiento candidato de TMVP, el MV coubicado puede necesitar ser escalado para compensar las diferencias de distancia temporal, como se muestra en la FIG. 11B.

[0122] Varios otros aspectos de los modos de fusión y AMVP son dignos de mención como sigue. Por ejemplo, el codificador de video 200 y el decodificador de video 300 pueden realizar una escala de vector 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 lo contiene). Cuando se utiliza un vector de movimiento para predecir el otro vector de movimiento, la distancia de la imagen contenedora y la imagen de referencia se calcula sobre la base de los valores del recuento de orden de imágenes (POC).

[0123] Para predecir un vector de movimiento, tanto su imagen de contención asociada como su 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 función de estas dos distancias POC. Para un candidato vecino espacial, las imágenes de contención para los dos vectores de movimiento son las mismas, mientras que las imágenes de referencia son diferentes. En HEVC, la escala del vector de movimiento se aplica tanto a TMVP como a AMVP para candidatos vecinos espaciales y temporales.

[0124] Como otro ejemplo, el codificador de video 200 y el decodificador de video 300 pueden realizar una generación de candidatos de vectores de movimiento artificial. Si una lista de candidatos de vector de movimiento no está completa, los candidatos de vector de movimiento artificial se generan y se insertan al final de la lista hasta que tenga todos los candidatos.

[0125] En el modo de fusión, hay dos tipos de candidatos MV artificiales: candidato combinado derivado 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 de una combinación del vector de movimiento del primer candidato que se refiere a una imagen de la lista 0 y el vector de movimiento de un segundo candidato refiriéndose a una imagen de la lista 1.

[0126] Como otro ejemplo, el codificador de video 200 y el decodificador de video 300 pueden realizar un proceso de poda para la inserción de candidatos. Es posible que los candidatos de diferentes bloques sean los mismos, lo que reduce la eficacia de una lista de candidatos de fusión/AMVP. Se aplica un proceso de poda para solucionar este problema. Compara un candidato con los demás en la lista de candidatos actual para evitar insertar candidatos idénticos 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 anterior se compara con el candidato de fusión izquierdo, el candidato de fusión derecho anterior se compara con el candidato de fusión anterior, el candidato de fusión de abajo a la izquierda se compara con el candidato de fusión de izquierda, el candidato de fusión de izquierda arriba se compara con el candidato de fusión de izquierda y el candidato de fusión de arriba.

[0127] El codificador de video 200 y el decodificador de video 300 también pueden usar otros métodos de predicción de movimiento. En el desarrollo de la codificación de video versátil (VVC), se propuso un método 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 método 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 inmediatamente adyacentes. Se mantiene una tabla con múltiples candidatos 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 en una forma de primero en entrar, primero en salir (FIFO) como un nuevo candidato de HMVP. Entonces, se puede aplicar una regla de restricción FIFO. Al insertar un HMVP en la tabla, se puede aplicar primero una verificación de redundancia para encontrar si hay un HMVP idéntico en la tabla. Si se encuentra, ese HMVP en particular puede eliminarse de la tabla y, posteriormente, todos los candidatos HMVP se mueven.

[0128] Se pueden usar candidatos de HMVP en el proceso de construcción de la lista de candidatos de fusión. Por ejemplo, todos los candidatos HMVP desde la última entrada hasta la primera entrada en la tabla pueden insertarse después del candidato TMVP. La poda se puede aplicar 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 para fusión finaliza.

[0129] De manera similar, los candidatos de HMVP también se pueden usar en el proceso de construcción de la lista de candidatos de AMVP. Los vectores de movimiento de los últimos candidatos K HMVP en la tabla se pueden insertar después del candidato TMVP. En algunos ejemplos, solo se utilizan candidatos HMVP con la misma imagen de referencia que la imagen de referencia objetivo AMVP para construir la lista de candidatos AMVP. La poda se puede aplicar a los candidatos HMVP.

[0130] En HEVC, la codificación de una CTU actual solo puede depender de las CTU a la izquierda, arriba a la izquierda, arriba y arriba a la derecha. Por lo tanto, el procesamiento paralelo de frente de onda (WPP) se puede admitir en HEVC. Sin embargo, el método HMVP en K0104 puede causar dependencia entre un bloque actual y todas las CTU codificadas previamente en un segmento. Por lo tanto, es posible que WPP no se aplique si se utiliza el método HMVP. Esta divulgación describe técnicas para usar HMVP con inicialización CTU, en las que las dependencias siguen siendo las mismas que en HEVC. Esta divulgación también describe técnicas para HMVP con inicialización de fila CTU (reinicio).

[0131] Según las técnicas de esta divulgación, el codificador de video 200 y el

decodificador de video 300 pueden realizar HMVP con inicialización de CTU. La tabla HMVP se inicializa al comienzo de cada CTU. La inicialización puede agregar MV de los bloques codificados vecinos inmediatos de la CTU actual en la tabla HMVP. El bloque codificado vecino inmediato puede estar a la izquierda, arriba, arriba a la izquierda o arriba a la derecha de la CTU actual como en HEVC. Si la predicción del vector de movimiento temporal está habilitada, el bloque codificado vecino inmediato también puede ser el bloque coubicado en la imagen de referencia.

[0132] La FIG. 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 utilizan para inicializar la tabla HMVP. En la Figura 1 se muestra un ejemplo del uso de candidatos de fusión temporal y espacial HEVC. 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". Tener en cuenta que el candidato de fusión temporal inferior derecho y el candidato inferior izquierdo (3) no están disponibles ya que sus ubicaciones están por debajo de la línea CTU actual.

[0133] En otro ejemplo, el proceso de derivación del candidato de fusión para el bloque CTU actual se usa para inicializar la tabla HMVP. Además de los candidatos de fusión espacial y temporal, también se pueden utilizar otros candidatos de fusión (por ejemplo, candidatos de vector de movimiento artificial) para la inicialización.

[0134] La FIG. 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 codificar la primera CU, los candidatos de fusión espacial y temporal de la primera CU, como se muestra en la FIG. 13, se agregan a la tabla HMVP. Y luego, el MV de la primera CU también se agrega si la CU está codificada interpredicción. Se debe tener en cuenta que, si la primera CU no es igual a la CTU actual, se pueden agregar dos candidatos de fusión temporal "T0" y "T1" en orden. La FIG. 13 muestra un ejemplo de fusión de candidatos de la primera UC en una CTU.

[0135] En otro ejemplo, la tabla HMVP se inicializa como vacía al comienzo de la CTU actual de codificación. Sin embargo, después de codificar la primera CU, todos los candidatos de fusión de la primera CU se agregan a la tabla HMVP. Y luego también se agrega el MV de la primera CU si está codificado interpredicción.

[0136] El codificador de video 200 y el decodificador de video 300 también pueden realizar HMVP con inicialización de fila CTU. En otro ejemplo, la inicialización de CTU

para HMVP descrita anteriormente se aplica solo en la primera CTU en una fila de CTU. De manera similar al HMVP en K0104, el proceso de poda se puede aplicar a la tabla inicializada para eliminar algunos o todos los duplicados. El proceso de poda tampoco se puede aplicar a la tabla inicializada para reducir la complejidad.

[0137] La FIG. 14 es un diagrama de bloques que ilustra un codificador de video de ejemplo 200 que puede realizar las técnicas de la presente divulgación. Se proporciona con fines de explicación y no debe considerarse que limita las técnicas como se ejemplifica y describe ampliamente en esta divulgación. Para fines de explicación, esta divulgación describe el codificador de video 200 en el contexto de estándares de codificación de video tales como el estándar de codificación de video HEVC y el estándar de codificación de video H.266 en desarrollo. Sin embargo, las técnicas de la presente divulgación no se limitan a estos estándares de codificación de video, y son aplicables generalmente a la codificación y decodificación de video.

[0138] En el ejemplo de la FIG. 14, el codificador de video 200 incluye memoria de datos de video 230, unidad de selección de modo 202, unidad de generación residual 204, unidad de procesamiento de transformación 206, unidad de cuantificación 208, unidad de cuantificación inversa 210, unidad de procesamiento de transformada inversa 212, unidad de reconstrucción 214, unidad de filtro 216, búfer de imagen decodificada (DPB) 218, y unidad de codificación de entropía 220. Cualquiera o la totalidad de la memoria de datos de video 230, unidad de selección de modo 202, unidad de generación residual 204, unidad de procesamiento de transformación 206, unidad de cuantificación 208, unidad de cuantificación inversa 210, unidad de procesamiento de transformada 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 video 200 puede incluir procesadores adicionales o alternativos o circuitos de procesamiento para realizar estas y otras funciones.

[0139] La memoria de datos de video 230 puede almacenar datos de video para ser codificados por los componentes del codificador de video 200. El codificador de video 200 puede recibir los datos de video almacenados en la memoria de datos de video 230 de, por ejemplo, la fuente de video 104 (FIG. 1). El DPB 218 puede actuar como una memoria de imagen de referencia que almacena datos de video de referencia para su uso en la predicción de datos de video posteriores por el codificador de video 200. La memoria de datos de video 230 y el DPB 218 pueden estar formados por cualquiera de una variedad de dispositivos de memoria, como la dinámica aleatoria memoria de

acceso (DRAM), que incluye DRAM síncrona (SDRAM), RAM magnetorresistiva (MRAM), RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. La memoria de datos de video 230 y el DPB 218 pueden ser proporcionados por el mismo dispositivo de memoria o dispositivos de memoria separados. En varios ejemplos, la memoria de datos de video 230 puede estar en chip con otros componentes del codificador de video 200, como se ilustra, o fuera del chip en relación con esos componentes.

[0140] En esta divulgación, la referencia a la memoria de datos de video 230 no debe interpretarse como limitada a la memoria interna del codificador de video 200, a menos que se describa específicamente como tal, o la memoria externa al codificador de video 200, a menos que se describa específicamente como tal. Más bien, la referencia a la memoria de datos de video 230 debe entenderse como memoria de referencia que almacena datos de video que el codificador de video 200 recibe para codificar (por ejemplo, datos de video para un bloque actual que se codificará). La memoria 106 de la FIG. 1 también puede proporcionar almacenamiento temporal de salidas de las diversas unidades del codificador de video 200.

[0141] Las diversas unidades de la FIG. 14 se ilustran para ayudar a comprender las operaciones realizadas por el codificador de video 200. Las unidades pueden implementarse como circuitos de función fija, circuitos programables o una combinación de los mismos. Los circuitos de función fija se refieren a circuitos que proporcionan una funcionalidad particular y están preestablecidos en las operaciones que se pueden realizar. Los circuitos programables se refieren a circuitos que se pueden programar para realizar diversas tareas y proporcionan una funcionalidad flexible en las operaciones que se pueden realizar. Por ejemplo, los circuitos programables pueden ejecutar *software* o *firmware* que hacen que los circuitos programables funcionen de la manera definida por las instrucciones del *software* o *firmware*. 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 unidades pueden ser bloques de circuitos distintos (función fija o programable), y en algunos ejemplos, la una o más unidades pueden ser circuitos integrados.

[0142] El codificador de video 200 puede incluir unidades lógicas aritméticas (ALU), unidades de funciones elementales (EFU), circuitos digitales, circuitos analógicos y/o núcleos programables, formados a partir de circuitos programables. En los ejemplos en los que las operaciones del codificador de video 200 se realizan utilizando un *software* ejecutado por los circuitos programables, la memoria 106 (FIG. 1) puede almacenar el

código objeto del *software* que el codificador de video 200 recibe y ejecuta, u otra memoria dentro del codificador de video 200 (no mostrado) puede almacenar tales instrucciones.

[0143] La memoria de datos de video 230 está configurada para almacenar datos de video recibidos. El codificador de video 200 puede recuperar una imagen de los datos de video de la memoria de datos de video 230 y proporcionar los datos de video a la unidad de generación residual 204 y la unidad de selección de modo 202. Los datos de video en la memoria de datos de video 230 pueden ser datos de video sin procesar que se codificarán.

[0144] 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 predicciones de video de acuerdo con otros modos de predicción. Como ejemplos, la unidad de selección de modo 202 puede incluir una unidad 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 similar.

[0145] La unidad de selección de modo 202 generalmente coordina múltiples pases de codificación para probar combinaciones de parámetros de codificación y los valores de distorsión de velocidad resultantes para tales combinaciones. Los parámetros de codificación pueden incluir la división de CTU en CU, modos de predicción para las CU, tipos de transformación para datos residuales de las CU, parámetros de cuantificación para datos residuales de las CU, etc. La unidad de selección de modo 202 puede finalmente seleccionar la combinación de parámetros de codificación que tienen valores de distorsión de velocidad que son mejores que las otras combinaciones probadas.

[0146] El codificador de video 200 puede dividir una imagen recuperada de la memoria de datos de video 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 quadtree de HEVC descrita con anterioridad. Como se describió con anterioridad, el codificador de video 200 puede formar una o más CU de particionar una CTU de acuerdo con la estructura de árbol. Tal CU también se puede denominar generalmente como un “bloque de video” o “bloque”.

[0147] En general, la unidad de selección de modo 202 también controla sus componentes (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 parte 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 coincidan estrechamente en una o más imágenes de referencia (por ejemplo, una o más imágenes codificadas previamente almacenadas en el 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 cuadradas (SSD), la diferencia absoluta media (MAD), diferencias cuadráticas medias (MSD) o similares. La unidad de estimación de movimiento 222 generalmente puede realizar estos cálculos usando diferencias muestra por muestra entre el bloque actual y el bloque de referencia que se está considerando. 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.

[0148] 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 respecto a la posición del bloque actual en una imagen actual. La unidad de estimación de movimiento 222 puede proporcionar los vectores de movimiento a la unidad de compensación de movimiento 224. Por ejemplo, para la predicción 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 entonces un bloque de predicción usando los vectores de movimiento. Por ejemplo, la unidad de compensación de movimiento 224 puede recuperar datos del bloque de referencia usando el vector de movimiento. Como otro ejemplo, si el vector de movimiento tiene una precisión de muestra fraccional, 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 los respectivos vectores de movimiento y combinar los datos recuperados, por ejemplo, el promedio de muestra por muestra o el promedio ponderado.

[0149] De acuerdo con las técnicas de esta divulgación, el búfer de imagen decodificada 218 puede incluir uno o más búferes MVP históricos para líneas de CTU. Es decir, a cada línea CTU se le puede asignar su propio búfer MVP, o se puede usar un único búfer MVP para una pluralidad de líneas CTU. En cualquier caso, el codificador de video 200 puede restablecer un búfer MVP para una línea CTU al comienzo de la decodificación de datos de video de la línea CTU. La unidad de compensación de movimiento 224 u otra unidad de codificador de video 200 puede configurarse para almacenar sólo vectores de movimiento únicos en el búfer MVP. Como se discutió anteriormente, la unidad de compensación de movimiento 224 u otra unidad de codificador de video 200 puede configurarse para usar una regla FIFO para administrar la información de movimiento almacenada en el búfer MVP, de modo que al agregar un vector de movimiento al búfer MVP, si el búfer MVP es la unidad 224 de compensación de movimiento completa puede eliminar un vector de movimiento agregado más temprano del búfer MVP. En algunos ejemplos, el codificador de video 200 puede mantener diferentes búferes MVP respectivos para cada uno de una variedad de modelos de movimiento, tales como, por ejemplo, modelo de movimiento afín, información de movimiento del modo de copia intrabloque, información de movimiento de compensación de iluminación local, sub-bloque MVP y predicción del movimiento temporal.

[0150] 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 modos direccionales, la unidad de intrapredicción 226 generalmente puede combinar matemáticamente valores de muestras vecinas y llenar 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 intrapredicción 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.

[0151] 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 del bloque actual desde la memoria de datos de video 230 y el bloque de predicción desde la unidad de selección de modo 202. Unidad de generación residual 204 calcula las diferencias muestra por muestra entre el bloque actual y el bloque de predicción. Las diferencias de muestra por muestra resultantes definen un bloque residual para el bloque actual. En algunos ejemplos, la unidad de generación

residual 204 también puede determinar las diferencias entre los valores de muestra en el bloque residual para generar un bloque residual utilizando la modulación de código de pulso diferencial residual (RDPCM). En algunos ejemplos, la unidad de generación residual 204 puede formarse usando uno o más circuitos sustractores que realizan sustracción binaria.

[0152] En los ejemplos en los que la unidad de selección de modo 202 divide las CU en PU, cada PU puede asociarse con una unidad de predicción de luma y las unidades de predicción de croma correspondientes. El codificador de video 200 y el decodificador de video 300 pueden soportar PU que tienen varios tamaños. Como se indicó con anterioridad, el tamaño de una CU puede referirse al tamaño del bloque de codificación 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 video 200 puede admitir tamaños de PU de $2N \times 2N$ o $N \times N$ para intrapredicción, y tamaños de PU simétricos de $2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$ o similares para interpredicción. El codificador de video 200 y el decodificador de video 300 también pueden admitir particiones asimétricas para tamaños de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$ y $nR \times 2N$ para interpredicción.

[0153] En los ejemplos en los que la unidad de selección de modo no divide más una CU en PU, cada CU puede estar asociada con un bloque de codificación luma y los bloques de codificación croma correspondientes. Como se indicó con anterioridad, el tamaño de una CU puede referirse al tamaño del bloque de codificación luma de la CU. El codificador de video 200 y el decodificador de video 120 pueden admitir tamaños CU de $2N \times 2N$, $2N \times N$ o $N \times 2N$.

[0154] Para otras técnicas de codificación de video tales como una codificación en modo de copia intrabloque, una codificación en modo afín y una codificación en modo de modelo lineal (LM), como algunos ejemplos, la unidad de selección de modo 202, a través de las unidades respectivas asociadas con la codificación técnicas, genera un bloque de predicción para el bloque actual que se codifica. En algunos ejemplos, 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 de 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 de entropía 220 a codificar.

[0155] Como se describió con anterioridad, la unidad de generación residual 204 recibe los datos de video 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 por muestra entre el bloque de predicción y el bloque actual.

[0156] 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 este documento “bloque de coeficiente de transformación”). La unidad de procesamiento de transformación 206 puede aplicar diversas transformaciones a un bloque residual para formar el bloque de coeficiente de transformación. Por ejemplo, la unidad de procesamiento de transformación 206 puede aplicar una transformada de coseno discreto (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 transformadas 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 transformadas a un bloque residual.

[0157] La unidad de cuantificación 208 puede cuantificar los coeficientes de transformación en un bloque de coeficiente de transformación, para producir un bloque de coeficiente de transformación cuantificado. La unidad de cuantificación 208 puede cuantificar los coeficientes de transformación de un bloque de coeficiente de transformación de acuerdo con un valor de parámetro de cuantificación (QP) asociado con el bloque actual. El codificador de video 200 (por ejemplo, a través de la unidad de selección de modo 202) puede ajustar el grado de cuantificación aplicado a los bloques de coeficientes de transformación asociados con el bloque actual ajustando el 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.

[0158] La unidad de cuantificación inversa 210 y la unidad de procesamiento de transformación inversa 212 pueden aplicar cuantificación inversa y transformadas inversas a un bloque de coeficiente de transformación cuantificado, respectivamente, para reconstruir un bloque residual del bloque de coeficiente de transformación. La unidad de reconstrucción 214 puede producir un bloque reconstruido correspondiente al bloque actual (aunque potencialmente con cierto grado de distorsión) basado en el bloque residual reconstruido y un bloque de predicción generado por la unidad de

selección de modo 202. Por ejemplo, la unidad de reconstrucción 214 puede agregar 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.

[0159] 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 filtro 216 pueden omitirse, en algunos ejemplos.

[0160] El codificador de video 200 almacena bloques reconstruidos en el DPB 218. Por ejemplo, en ejemplos donde no se necesitan 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 operaciones de la unidad de filtro 216, la unidad de filtro 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 los bloques de imágenes codificadas posteriormente. 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.

[0161] En general, la unidad de codificación de entropía 220 puede codificar entropía elementos de sintaxis recibidos de otros componentes funcionales del codificador de video 200. Por ejemplo, la unidad de codificación de entropía 220 puede codificar entropía bloques de coeficientes de transformación cuantificados de la unidad de cuantificación 208. Como otro ejemplo, codificación de entropía la unidad 220 puede codificar entropía elementos de sintaxis de predicción (por ejemplo, información de movimiento para interpredicción o información intramodal para intrapredicción) desde la unidad de selección de modo 202. La unidad 220 de codificación de entropía puede realizar una o más operaciones de codificación de entropía en los elementos de sintaxis, que son otro ejemplo de datos de video, para generar datos codificados por entropía. Por ejemplo, la unidad de codificación de entropía 220 puede realizar una operación de codificación de longitud variable adaptativa al contexto (CAVLC), una operación de CABAC, una operación de codificación de longitud variable a variable (V2V), una codificación aritmética binaria adaptativa al contexto basada en sintaxis (SBAC)), una operación de codificación de Entropía de Partición de Intervalo de Probabilidad (PIPE), una operación de codificación exponencial-Golomb u otro tipo de operación de

codificación de entropía en los datos. En algunos ejemplos, la unidad de codificación de entropía 220 puede funcionar en modo de derivación donde los elementos de sintaxis no están codificados por entropía.

[0162] El codificador de video 200 puede emitir un flujo de bits que incluye los elementos de sintaxis codificados en entropía necesarios para reconstruir bloques de una porción o imagen. En particular, la unidad de codificación de entropía 220 puede emitir el flujo de bits

[0163] Las operaciones descritas con anterioridad se describen con respecto a un bloque. Dicha descripción debe entenderse como operaciones para un bloque de codificación luma y/o bloques de codificación croma. Como se describió con anterioridad, en algunos ejemplos, el bloque de codificación luma y los bloques de codificación croma son componentes de luma y croma de un CU. En algunos ejemplos, el bloque de codificación luma y los bloques de codificación croma son componentes de luma y croma de una PU.

[0164] 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 un ejemplo, las operaciones para identificar un vector de movimiento (MV) y una imagen de referencia para un bloque de codificación luma no necesitan repetirse para identificar un MV y una imagen de referencia para los bloques de croma. Por el contrario, el MV para el bloque de codificación luma se puede escalar 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 los bloques de codificación de luma y los bloques de codificación de croma.

[0165] El codificador de video 200 representa un ejemplo de un dispositivo configurado para codificar datos de video que incluye una memoria configurada para almacenar datos de video 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 un primer búfer de predictor de vector de movimiento histórico (MVP) de la memoria, y almacenar información de movimiento para una segunda línea CTU en un segundo búfer MVP histórico de la memoria, siendo la segunda línea CTU diferente de la primera línea CTU.

[0166] El codificador de video 200 también representa un ejemplo de un dispositivo configurado para codificar datos de video que incluye una memoria configurada para almacenar datos de video, y una o más unidades de procesamiento implementadas en circuitos y configuradas para almacenar información de movimiento codificado en un

predicador de vector de movimiento histórico (MVP), almacena un tipo diferente de información de movimiento en el búfer del historial MVP y codifica la información de movimiento de un bloque de datos de video utilizando la información de movimiento del búfer del historial MVP.

[0167] El codificador de video 200 también representa un ejemplo de un dispositivo configurado para codificar datos de video que incluye una memoria configurada para almacenar datos de video, y una o más unidades de procesamiento implementadas en circuitos y configuradas para almacenar una pluralidad de diferentes tipos de información de movimiento en respectivas, búferes de predicción de vector de movimiento (MVP) histórico diferente.

[0168] La FIG. 15 es un diagrama de bloques que ilustra un decodificador de video 300 de ejemplo que puede realizar las técnicas de la presente divulgación. La FIG. 15 se proporciona con fines de explicación y no limita las técnicas como se ejemplifica y describe ampliamente en esta divulgación. Para fines de explicación, esta divulgación describe el decodificador de video 300 de acuerdo con las técnicas de JEM y HEVC. Sin embargo, las técnicas de la presente divulgación pueden realizarse mediante dispositivos de codificación de video que están configurados de acuerdo con otros estándares de codificación de video.

[0169] En el ejemplo de la FIG. 15, el decodificador de video 300 incluye memoria de búfer de imagen codificada (CPB) 320, unidad de decodificación de entropía 302, unidad de procesamiento de predicción 304, unidad de cuantificación inversa 306, unidad de procesamiento de transformada inversa 308, unidad de reconstrucción 310, unidad de filtro 312 y búfer de imagen decodificada (DPB) 314. Cualquiera o la totalidad de la memoria 320 de CPB, 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 el DPB 314 pueden implementarse en uno o más procesadores o en circuitos de procesamiento. Además, el decodificador de video 300 puede incluir procesadores o circuitos de procesamiento adicionales o alternativos para realizar estas y otras funciones.

[0170] 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 afín, una unidad de modelo lineal (LM) o similares. En otros ejemplos, el decodificador de video 300 puede incluir más, menos o componentes funcionales diferentes.

[0171] La memoria 320 de CPB puede almacenar datos de video, tales como un flujo de bits de video codificado, para ser decodificados por los componentes del decodificador de video 300. Los datos de video almacenados en la memoria 320 de CPB pueden obtenerse, por ejemplo, del medio 110 legible por computadora (FIG. 1). La memoria 320 de CPB puede incluir un CPB que almacena datos de video codificados (por ejemplo, elementos de sintaxis) de un flujo de bits de video codificado. Además, la memoria CPB 320 puede almacenar datos de video que no sean elementos de sintaxis de una imagen codificada, como datos temporales que representan salidas de las diversas unidades del decodificador de video 300. El DPB 314 generalmente almacena imágenes decodificadas, que el decodificador de video 300 puede generar y/o usar como datos de video de referencia al decodificar datos o imágenes posteriores del flujo de bits de video codificado. La memoria CPB 320 y el DPB 314 pueden estar formadas por cualquiera de una variedad de dispositivos de memoria, como la memoria de acceso aleatorio dinámico (DRAM), incluida 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 ser proporcionadas por el mismo dispositivo de memoria o dispositivos de memoria separados. En varios ejemplos, la memoria 320 de CPB puede estar en chip con otros componentes del decodificador de video 300, o fuera de chip en relación con esos componentes.

[0172] Adicional o alternativamente, en algunos ejemplos, el decodificador de video 300 puede recuperar datos de video codificados de la memoria 120 (FIG. 1). Es decir, la memoria 120 puede almacenar datos como se discutió con anterioridad con la memoria 320 de CPB. Del mismo modo, la memoria 120 puede almacenar instrucciones para ser ejecutadas por el decodificador de video 300, cuando parte o la totalidad de la funcionalidad del decodificador de video 300 se implementa en el *software* para ser ejecutado por circuitos de procesamiento del decodificador de video 300.

[0173] Las diversas unidades mostradas en la FIG. 15 se ilustran para ayudar a comprender las operaciones realizadas por el decodificador de video 300. Las unidades pueden implementarse como circuitos de función fija, circuitos programables o una combinación de los mismos. Similar a la FIG. 14, los circuitos de función fija se refieren a circuitos que proporcionan una funcionalidad particular, y están preestablecidos en las

operaciones que se pueden realizar. Los circuitos programables se refieren a circuitos que se pueden programar para realizar diversas tareas y proporcionan una funcionalidad flexible en las operaciones que se pueden realizar. Por ejemplo, los circuitos programables pueden ejecutar *software* o *firmware* que hacen que los circuitos programables funcionen de la manera definida por las instrucciones del *software* o *firmware*. 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 (función fija o programable), y en algunos ejemplos, la una o más unidades pueden ser circuitos integrados.

[0174] El decodificador de video 300 puede incluir ALU, EFU, circuitos digitales, circuitos analógicos y/o núcleos programables formados a partir de circuitos programables. En los ejemplos en los que las operaciones del decodificador de video 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 de objeto) del *software* que el decodificador de video 300 recibe y ejecuta.

[0175] La unidad de decodificación de entropía 302 puede recibir datos de video codificados del CPB y decodificar por entropía los datos de video 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 datos de video decodificados basados en los elementos de sintaxis extraídos del flujo de bits.

[0176] En general, el decodificador de video 300 reconstruye una imagen bloque por bloque. El decodificador de video 300 puede realizar una operación de reconstrucción en cada bloque individualmente (donde el bloque que se está reconstruyendo actualmente, es decir, decodificado, puede denominarse un “bloque actual”).

[0177] La unidad de decodificación de entropía 302 puede decodificar por entropía elementos de sintaxis que definen coeficientes de transformación cuantificados de un bloque de coeficiente de transformación cuantificado, así como información de transformación, tal como un parámetro de cuantificación (QP) y/o indicaciones 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, asimismo, un grado de cuantificación inversa para aplicar la unidad de cuantificación inversa 306. La unidad de cuantificación inversa 306 puede, por

ejemplo, realizar una operación de desplazamiento a la izquierda bit a bit para cuantificar inversamente los coeficientes de transformación cuantificados. La unidad de cuantificación inversa 306 puede formar así un bloque de coeficiente de transformación que incluye coeficientes de transformación.

[0178] Después de que la unidad de cuantificación inversa 306 forma el bloque de coeficiente de transformación, la unidad de procesamiento de transformación inversa 308 puede aplicar una o más transformaciones inversas al bloque de coeficiente 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 transformada entera inversa, una transformada de Karhunen-Loeve inversa (KLT), una transformada rotacional inversa, una transformada direccional inversa u otra transformación inversa al bloque de coeficientes.

[0179] Además, la unidad de procesamiento de predicción 304 genera un bloque de predicción de acuerdo con los elementos de sintaxis de información de predicción que fueron decodificados por entropía por la unidad de decodificación de entropía 302. Por ejemplo, si los elementos de sintaxis de información de predicción indican que el bloque actual es interpredicho, compensación de movimiento la unidad 316 puede generar el bloque de predicción. En este caso, los elementos de sintaxis de información de predicción pueden indicar una imagen de referencia en el DPB 314 desde la cual recuperar un bloque de referencia, así como un vector de movimiento que identifica una ubicación del bloque de referencia en la imagen de referencia en relación con 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 (FIG. 14).

[0180] De acuerdo con las técnicas de esta divulgación, el búfer de imagen decodificada 314 puede incluir uno o más búferes MVP históricos para líneas de CTU. Es decir, a cada línea CTU se le puede asignar su propio búfer MVP, o se puede usar un único búfer MVP para una pluralidad de líneas CTU. En cualquier caso, el decodificador 300 de video puede restablecer un búfer MVP para una línea CTU al comienzo de la codificación de datos de video de la línea CTU. La unidad de compensación de movimiento 316 u otra unidad de decodificador de video 300 puede configurarse para almacenar solo vectores de movimiento únicos en el búfer MVP. Como se discutió anteriormente, la unidad de compensación de movimiento 316 u otra unidad de decodificador de video 300 puede configurarse para usar una regla FIFO para

administrar la información de movimiento almacenada en el búfer MVP, de modo que al agregar un vector de movimiento al búfer MVP, si el búfer MVP está la unidad de compensación de movimiento completa 316 puede eliminar un vector de movimiento agregado más temprano del búfer MVP. En algunos ejemplos, el decodificador de video 300 puede mantener diferentes búferes MVP respectivos para cada uno de una variedad de modelos de movimiento, como, por ejemplo, modelo de movimiento afín, información de movimiento del modo de copia intrabloque, información de movimiento de compensación de iluminación local, sub-bloque MVP y predicción del movimiento temporal.

[0181] Como otro ejemplo, si los elementos de sintaxis de información de predicción indican que el bloque actual es intrapredicho, la unidad de interpredicción 318 puede generar el bloque de predicción de acuerdo con un modo de intrapredicción indicado por los elementos de sintaxis de información de predicción. Nuevamente, la unidad de intrapredicción 318 generalmente puede realizar el proceso de intrapredicción de una manera que es sustancialmente similar a la descrita con respecto a la unidad de intrapredicción 226 (FIG. 14). La unidad de interpredicción 318 puede recuperar datos de muestras vecinas al bloque actual del DPB 314.

[0182] La unidad de reconstrucción 310 puede reconstruir el bloque actual usando el bloque de predicción y el bloque residual. Por ejemplo, la unidad de reconstrucción 310 puede agregar muestras del bloque residual a las muestras correspondientes del bloque de predicción para reconstruir el bloque actual.

[0183] 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 filtro 312 no se realizan necesariamente en todos los ejemplos.

[0184] El decodificador de video 300 puede almacenar los bloques reconstruidos en el DPB 314. Por ejemplo, en ejemplos donde no son necesarias operaciones de la unidad de filtro 312, la unidad de reconstrucción 310 puede almacenar bloques reconstruidos en el DPB 314. En ejemplos donde las operaciones de la unidad de filtro 312 son necesarias, la unidad de filtro 312 puede almacenar los bloques reconstruidos filtrados en el DPB 314. Como se discutió anteriormente, el DPB 314 puede proporcionar información de referencia, tal como muestras de una imagen actual para intrapredicción e imágenes decodificadas previamente para compensación de movimiento posterior, a la unidad de procesamiento de predicción 304. Además, el decodificador de video 300

puede emitir imágenes decodificadas desde el DPB 314 para su posterior presentación en un dispositivo de visualización, tal como el dispositivo de visualización 118 de la FIG. 1.

[0185] El decodificador de video 300 representa un ejemplo de un dispositivo configurado para decodificar datos de video que incluye una memoria configurada para almacenar datos de video 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 ser procesada por un primer subproceso de un proceso de codificación de video) en un primer búfer de predictor de vector de movimiento histórico (MVP) de la memoria, y almacenar información de movimiento para una segunda línea CTU (que puede ser procesada por un segundo subproceso del proceso de codificación de video) en un segundo búfer MVP histórico de la memoria, siendo la segunda línea CTU diferente a la primera línea CTU. El segundo subproceso puede ser diferente al primer subproceso.

[0186] El decodificador 300 de video también representa un ejemplo de un dispositivo configurado para decodificar datos de video que incluye una memoria configurada para almacenar datos de video, y una o más unidades de procesamiento implementadas en circuitos y configuradas para almacenar información de movimiento codificado en un predictor de vector de movimiento histórico (MVP), almacena un tipo diferente de información de movimiento en el búfer del historial MVP y codifica la información de movimiento de un bloque de datos de video utilizando la información de movimiento del búfer histórico del MVP.

[0187] El decodificador 300 de video también representa un ejemplo de un dispositivo configurado para decodificar datos de video que incluye una memoria configurada para almacenar datos de video, y una o más unidades de procesamiento implementadas en circuitos y configuradas para almacenar una pluralidad de diferentes tipos de información de movimiento en respectivas, búferes históricos de predicción de vector de movimiento (MVP) diferentes.

[0188] La FIG. 16 es un diagrama de flujo que ilustra un método de ejemplo para codificar un bloque actual de acuerdo con las técnicas de esta divulgación. El bloque actual puede comprender una CU actual. Aunque se describe con respecto al codificador de video 200 (FIGS. 1 y 14), debe entenderse que otros dispositivos pueden configurarse para realizar un método similar al de la FIG. 16.

[0189] En este ejemplo, el codificador de video 200 predice inicialmente el bloque actual usando información de movimiento (350). Por ejemplo, el codificador de video 200

puede formar un bloque de predicción para el bloque actual usando la información de movimiento. El codificador de video 200 puede entonces calcular un bloque residual para el bloque actual (352). Para calcular el bloque residual, el codificador de video 200 puede calcular una diferencia entre el bloque no codificado original y el bloque de predicción para el bloque actual. El codificador de video 200 puede transformar y cuantificar entonces los coeficientes del bloque residual (354). A continuación, el codificador de video 200 puede escanear los coeficientes de transformación cuantificados del bloque residual (356). Durante la exploración, o después de la exploración, el codificador de video 200 puede codificar entropía los coeficientes y la información de movimiento (358) utilizando las técnicas de esta divulgación. El codificador de video 200 puede codificar los coeficientes usando CAVLC o CABAC.

[0190] El codificador de video 200 puede construir una lista de candidatos de información de movimiento que incluye, por ejemplo, candidatos de HMVP, según cualquiera o todas las técnicas de esta divulgación, seleccionar un índice de candidato que represente un predictor para la información de movimiento del bloque y codificar entropía el índice de candidatos. De acuerdo con las técnicas de esta divulgación, el codificador de video 200 puede restablecer un búfer MVP antes de usar el búfer MVP para almacenar información de movimiento de una línea CTU correspondiente. En algunos ejemplos, cada línea CTU puede tener su propio búfer MVP, o puede usarse un búfer MVP para múltiples líneas CTU. Además, el codificador de video 200 puede almacenar múltiples tipos de información de movimiento en búferes MVP, por ejemplo, el mismo búfer o búferes de modelos de movimiento respectivos diferentes. El codificador de video 200 puede codificar información de movimiento de un bloque actual usando predictores de vector de movimiento seleccionados de los datos del búfer MVP. El codificador de video 200 puede entonces dar salida a los datos codificados de entropía del bloque (360), por ejemplo, incluyendo los datos para los coeficientes y la información de movimiento, tal como el índice candidato.

[0191] De esta manera, el método de la FIG. 16 representa un ejemplo de un método que incluye almacenar información de movimiento para una primera línea de unidad de árbol de codificación (CTU) de una imagen en un primer búfer de predicción de vector de movimiento histórico (MVP); restablecer un segundo búfer MVP histórico de la memoria; y después de restablecer el segundo búfer MVP histórico, almacenar información de movimiento para una segunda línea CTU de la imagen en el segundo búfer MVP histórico, siendo la segunda línea CTU diferente de la primera línea CTU.

[0192] El método de la FIG. 16 también representa un ejemplo de un método que incluye

almacenar información de movimiento en un búfer de predictor de vector de movimiento histórico (MVP); almacenar un tipo diferente de información de movimiento en el búfer del historial MVP; y codificar información de movimiento de un bloque de datos de video usando información de movimiento del búfer histórico MVP.

[0193] El método de la FIG. 16 también representa un ejemplo de un método que incluye almacenar una pluralidad de diferentes tipos de información de movimiento en búferes respectivos, diferentes del predictor de vector de movimiento histórico (MVP).

[0194] La FIG. 17 es un diagrama de flujo que ilustra un método de ejemplo para decodificar un bloque actual de datos de video de acuerdo con las técnicas de esta divulgación. El bloque actual puede comprender una CU actual. Aunque se describe con respecto al decodificador de video 300 (FIGS. 1 y 15), debe entenderse que otros dispositivos pueden configurarse para realizar un método similar al de la FIG. 17.

[0195] El decodificador de video 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 los coeficientes de un bloque residual correspondiente al bloque actual (370). Como se discutió anteriormente, la información de predicción codificada en entropía puede incluir, por ejemplo, un índice de candidatos en una lista de candidatos, que puede incluir candidatos de HMVP de acuerdo con las técnicas de esta divulgación. El decodificador de video 300 puede decodificar por entropía los datos codificados por entropía para determinar la información de predicción para el bloque actual y para reproducir los coeficientes del bloque residual (372). El decodificador de video 300 puede predecir el bloque actual (374), por ejemplo, usando un modo de interpredicción como se indica en la información de predicción para el bloque actual, para calcular un bloque de predicción para el bloque actual.

[0196] En particular, el decodificador 300 de video puede construir una lista de candidatos que incluya los candidatos de HMVP como se discutió anteriormente, y luego determinar un candidato de la lista de candidatos para usar como predictor de vector de movimiento para el bloque actual usando el índice de candidato decodificado. De acuerdo con las técnicas de esta divulgación, el decodificador 300 de video puede restablecer un búfer MVP antes de usar el búfer MVP para almacenar información de movimiento de una línea CTU correspondiente. En algunos ejemplos, cada línea CTU puede tener su propio búfer MVP, o puede usarse un búfer MVP para múltiples líneas CTU. Además, el decodificador 300 de video puede almacenar múltiples tipos de información de movimiento en búferes MVP, por ejemplo, el mismo búfer o búferes de modelos de movimiento respectivos diferentes. El decodificador 300 de video puede

seleccionar usando predictores de vector de movimiento a partir de datos del búfer MVP.

[0197] El decodificador 300 de video puede reconstruir entonces un vector de movimiento para el bloque actual usando el predictor de vector de movimiento, luego predecir el bloque actual usando el vector de movimiento para generar un bloque de predicción. El decodificador de video 300 puede entonces escanear inversamente los coeficientes reproducidos (376), para crear un bloque de coeficientes de transformación cuantificados. El decodificador de video 300 puede cuantificar inversamente y transformar inversamente los coeficientes para producir un bloque residual (378). El decodificador de video 300 puede decodificar finalmente el bloque actual combinando el bloque de predicción y el bloque residual (380).

[0198] De esta manera, el método de la FIG. 17 representa un ejemplo de un método que incluye almacenar información de movimiento para una primera línea de unidad de árbol de codificación (CTU) de una imagen en un primer búfer de predicción de vector de movimiento histórico (MVP); restablecer un segundo búfer MVP histórico de la memoria; y después de restablecer el segundo búfer MVP histórico, almacenar información de movimiento para una segunda línea CTU de la imagen en el segundo búfer MVP histórico, siendo la segunda línea CTU diferente de la primera línea CTU.

[0199] El método de la FIG. 17 también representa un ejemplo de un método que incluye almacenar información de movimiento en un búfer de predictor de vector de movimiento histórico (MVP); almacenar un tipo diferente de información de movimiento en el búfer del historial MVP; y codificar información de movimiento de un bloque de datos de video usando información de movimiento del búfer histórico MVP.

[0200] El método de la FIG. 17 también representa un ejemplo de un método que incluye almacenar una pluralidad de diferentes tipos de información de movimiento en respectivos búferes de predicción de vector de movimiento histórico (MVP) diferentes.

[0201] La FIG. 18 es un diagrama de flujo que ilustra un método de ejemplo de codificación (codificación o decodificación) de datos de video de acuerdo con las técnicas de esta divulgación. Por ejemplo, el método de la FIG. 18 puede realizarse durante el paso 350 de la FIG. 16 o el paso 374 de la FIG. 17. Con fines de ejemplo y explicación, el método de la FIG. 18 se explica con respecto al decodificador de video 300, aunque el codificador de video 200 también puede realizar este o un método similar.

[0202] El decodificador 300 de video puede codificar bloques de una primera línea CTU de una imagen (390), por ejemplo, utilizando intrapredicción o interpredicción. El decodificador de video 300 almacena información de movimiento de una primera línea de unidad de árbol de codificación (CTU) de una imagen en un primer búfer (392), por

ejemplo, del DPB 314. El decodificador de video 300 puede usar la información de movimiento del primer búfer para codificar la información de movimiento utilizada durante la codificación interpredicción. En algunos ejemplos, un primer subproceso de un proceso de codificación de video realizado por el decodificador 300 de video puede codificar la primera línea CTU.

[0203] El decodificador 300 de video también puede restablecer un segundo búfer (394), por ejemplo, del DPB 314. El segundo búfer puede ser el mismo que el primer búfer o un búfer diferente. El decodificador 300 de video también puede codificar bloques de una segunda línea CTU (396). El decodificador 300 de video puede almacenar información de movimiento de la segunda línea CTU en el segundo búfer (398). En algunos ejemplos, un segundo subproceso del proceso de codificación de video realizado por el decodificador 300 de video puede codificar la segunda línea CTU, donde el segundo subproceso es diferente al primer subproceso.

[0204] De esta manera, el método de la FIG. 18 representa un ejemplo de un método que incluye almacenar información de movimiento para una primera línea de unidad de árbol de codificación (CTU) de una imagen en un primer búfer de predicción de vector de movimiento histórico (MVP); restablecer un segundo búfer MVP histórico de la memoria; y después de restablecer el segundo búfer MVP histórico, almacenar información de movimiento para una segunda línea CTU de la imagen en el segundo búfer MVP histórico, siendo la segunda línea CTU diferente de la primera línea CTU.

[0205] Debe reconocerse que, dependiendo del ejemplo, ciertos actos o eventos de cualquiera de las técnicas descritas en el presente documento pueden realizarse en una secuencia diferente, pueden agregarse, fusionarse u omitirse por completo (por ejemplo, no todos los actos descritos o eventos son necesarios para la práctica de las técnicas). Además, en ciertos ejemplos, los actos o eventos pueden realizarse simultáneamente, por ejemplo, a través de procesamiento de subprocesos múltiples, procesamiento de interrupción o procesadores múltiples, en lugar de secuencialmente.

[0206] En uno o más ejemplos, las funciones descritas pueden implementarse en *hardware*, *software*, *firmware* o cualquier combinación de los mismos. Si se implementa en *software*, las funciones pueden almacenarse o transmitirse como una o más instrucciones o códigos en un medio legible por computadora y ejecutarse por una unidad de procesamiento basada en *hardware*. Los medios legibles por computadora pueden incluir medios de almacenamiento legibles por computadora, que corresponden a un medio tangible como los medios de almacenamiento de datos o medios de comunicación que incluyen cualquier medio que facilite la transferencia de un programa

de computadora de un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por computadora generalmente pueden corresponder a (1) medios tangibles de almacenamiento legibles por computadora que no son transitorios o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que puedan acceder una o más computadoras o uno o más procesadores para recuperar instrucciones, códigos y/o estructuras de datos para la implementación de las técnicas descritas en esta divulgación. Un producto de programa de computadora puede incluir un medio legible por computadora.

[0207] A modo de ejemplo, y sin limitación, dichos medios de almacenamiento legibles por computadora pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento en disco óptico, almacenamiento en disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que puede usarse para almacenar el código de programa deseado en forma de instrucciones o estructuras de datos y al que puede acceder una computadora. Además, cualquier conexión se denomina correctamente un medio legible por computadora. Por ejemplo, si las instrucciones se transmiten desde un sitio web, un servidor u otra fuente remota utilizando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o tecnologías inalámbricas como infrarrojos, radio y microondas, cable coaxial, cable de fibra óptica, par trenzado, DSL o tecnologías inalámbricas como infrarrojos, radio y microondas están incluidos en la definición de medio. Sin embargo, debe entenderse que los medios de almacenamiento legibles por computadora y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que se dirigen a medios de almacenamiento tangibles no transitorios. Disco, como se usa en este documento, incluye disco compacto (CD), disco láser, disco óptico, disco versátil digital (DVD), disquete y disco Blu-ray, donde los discos generalmente reproducen datos magnéticamente, mientras que los discos reproducen datos ópticamente con láser. Las combinaciones de lo anterior también deben incluirse dentro del alcance de los medios legibles por computadora.

[0208] Las instrucciones pueden ser ejecutadas por uno o más procesadores, tales como uno o más procesadores de señal digital (DSP), microprocesadores de propósito general, circuitos integrados de aplicación específica (ASIC), arreglos de compuerta programables en campo (FPGA) u otro equivalente integrado o circuitos lógicos discretos. En consecuencia, los términos “procesador” y “circuitaría de procesamiento”, como se usan en el presente documento, puede referirse a cualquiera de las estructuras

anteriores o cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en este documento puede proporcionarse dentro de módulos dedicados de *hardware* y/o *software* configurados para codificación y decodificación, o incorporados en un códec combinado. Además, las técnicas podrían implementarse completamente en uno o más circuitos o elementos lógicos.

[0209] Las técnicas de la presente divulgación pueden implementarse en una amplia variedad de dispositivos o aparatos, incluyendo 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*. Por el contrario, como se describió con anterioridad, varias unidades pueden combinarse en una unidad de *hardware* de códec o proporcionarse mediante una colección de unidades de *hardware* interoperativas, que incluyen uno o más procesadores como se describe con anterioridad, junto con el *software* y/o *firmware* adecuado.

[0210] Se han descrito varios ejemplos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.