MÉTODO, APARATO Y MEDIO DE ALMACENAMIENTO PARA SEÑALIZACIÓN DEL PREDICTOR DEL VECTOR DE MOVIMIENTO DE LA UNIDAD DE SUBPREDICCIÓN

[01] Esta solicitud reivindica el beneficio de la Solicitud No Provisoria de los Estados Unidos No. 16/443.113, presentada el 17 de junio de 2019, que reivindica la prioridad de la Solicitud Provisoria de los Estados Unidos No. 62/687.052, presentada el 19 de junio de 2018, cuyo contenido completo se incorpora en la presente como referencia.

CAMPO TÉCNICO

[02] La presente divulgación se refiere a la codificación de video y decodificación de video.

ANTECEDENTES DE LA INVENCIÓN

[03] Se pueden incorporar capacidades de video digitales en un amplio rango de dispositivos, incluyendo televisores digitales, sistemas de transmisión directa digitales, sistemas de transmisión inalámbricos, asistentes digitales personales (PDA), computadoras portátiles o de escritorio, computadoras tipo tableta, 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 satélite, los denominados "*smart phones*", dispositivos de teleconferencia de video, dispositivos de transmisión de videos en directo, y similares. Los dispositivos de video digital implementan técnicas de codificación de video, tal como las descritas en los estándares definidos por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Part 10, Advanced Video Coding (AVC), el estándar de High Efficiency Video (HEVC), ITU-T H.265/High Efficiency Video Coding (HEVC) y extensiones de tales estándares. Los dispositivos de video pueden transmitir, recibir, codificar, decodificar, y/o almacenar información de video digital de manera más eficiente mediante la aplicación de tales técnicas de codificación de video.

[04] Las técnicas de codificación de video incluyen predicción espacial (intraimagen) y/o predicción temporal (entre imágenes) para reducir o eliminar la redundancia

inherente a las secuencias de video. Para codificar video con base en bloques, un plano de video (por ejemplo, una imagen de video o un segmento de una imagen de video) se puede particionar 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 plano intracodificado (I) de una imagen se codifican utilizando predicción espacial con respecto a muestras de referencia en bloques contiguos en la misma imagen. Los bloques de video en un plano intercodificado (P o B) de una imagen pueden utilizar predicción espacial con respecto a muestras de referencia en bloques contiguos 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

[05] En general, la presente divulgación se refiere al movimiento del vector de predicción en los códecs de video. Por ejemplo, un predictor del vector de movimiento se selecciona de manera adaptativa a partir de dos listas de candidatos de predicción de vectores de movimiento, que son derivadas. La primera lista incluye candidatos de predicción del vector de movimiento del nivel PU, la segunda lista incluye candidatos de predicción del vector de movimiento del nivel sub-PU.

[06] En una realización de ejemplo, se discute un método de decodificación de datos de video. El método incluye recibir datos de video codificados, analizar una bandera de movimiento de la unidad de subpredicción a partir de los datos de video codificados, en respuesta a determinar que la bandera de movimiento de la unidad de subpredicción está activa, derivar una lista de candidatos de predicción del movimiento del nivel de la unidad de subpredicción, en respuesta a determinar que la bandera de movimiento de la unidad de subpredicción no está activa, derivar una lista de candidatos de predicción del movimiento del nivel de la unidad de predicción, seleccionar un predictor del vector de movimiento de cualquiera de las listas, ya sea la

de candidatos de predicción del movimiento del nivel de la unidad de subpredicción o de la lista de candidatos de predicción del movimiento del nivel de la unidad de predicción, y decodificar los datos de video codificados mediante el predictor del vector de movimiento seleccionado. Los datos de video codificados incluyen un bloque actual y donde la lista de candidatos de predicción del movimiento del nivel de la unidad de subpredicción y la lista de candidatos de predicción del movimiento del nivel de la unidad de predicción se derivan de bloques contiguos del bloque actual. Los bloques contiguos son contiguos espaciales del bloque actual en una imagen actual o contiguos temporales del bloque actual en una imagen codificada previamente. La lista de candidatos de predicción del movimiento del nivel de la unidad de subpredicción o la lista de candidatos de predicción del movimiento del nivel de la unidad de predicción se ordena al menos parcialmente con base en ocurrencias de predicción de movimiento de los bloques contiguos. Los píxeles en una unidad de predicción comparten una primera información del vector de movimiento y los píxeles en una unidad de subpredicción comparten una segunda información del vector de movimiento, y donde la primera información del vector de movimiento o la segunda información del vector se determinan a partir del predictor del vector de movimiento seleccionado. La lista de candidatos del vector de movimiento del nivel de la unidad de predicción incluye al menos uno de los siguientes: candidatos contiguos espaciales y candidatos contiguos temporales. La lista de candidatos de predicción del movimiento del nivel de la unidad de subpredicción incluye al menos uno de los siguientes: predicción de vectores de movimiento afín, predicción del vector de movimiento temporal alternativo (ATMVP), predicción del vector de movimiento espacial-temporal (STMVP), predicción del vector de movimiento plano, y derivación del vector de movimiento emparejado por patrón (PMVD). El método incluye derivar un índice de candidato de fusión en respuesta a determinar que la bandera de movimiento de la unidad de subpredicción está activa, donde el índice de candidato de fusión especifica el predictor del vector de movimiento a seleccionar.

[07] En otra realización de ejemplo, se discute un aparato para decodificar datos de video. El aparato incluye una memoria para almacenar datos de video codificados y

recibidos, y un procesador. El procesador configurado para analizar una bandera de movimiento de la unidad de subpredicción a partir de los datos de video codificados, en respuesta a determinar que la bandera de movimiento de la unidad de subpredicción está activa, derivar una lista de candidatos de predicción del movimiento del nivel de la unidad de subpredicción, en respuesta a determinar que la bandera de movimiento de la unidad de subpredicción no está activa, derivar una lista de candidatos de predicción de movimiento del nivel de la unidad de predicción, seleccionar un predictor del vector de movimiento de cualquiera de las listas, ya sea la de candidatos de predicción del movimiento del nivel de la unidad de subpredicción o de la lista de candidatos de predicción del movimiento del nivel de la unidad de predicción, y decodificar los datos de video codificados mediante el predictor del vector de movimiento seleccionado. Los datos de video codificados incluyen un bloque actual y donde la lista de candidatos de predicción del movimiento del nivel de la unidad de subpredicción y la lista de candidatos de predicción del movimiento del nivel de la unidad de predicción se derivan de bloques contiguos del bloque actual. Los bloques contiguos son contiguos espaciales del bloque actual en una imagen actual o contiguos temporales del bloque actual en una imagen codificada previamente. La lista de candidatos de predicción del movimiento del nivel de la unidad de subpredicción o la lista de candidatos de predicción del movimiento del nivel de la unidad de predicción se ordena al menos parcialmente con base en ocurrencias de predicción de movimiento de los bloques contiguos. Los píxeles en una unidad de predicción comparten una primera información del vector de movimiento y los píxeles en una unidad de subpredicción comparten una segunda información del vector de movimiento, y donde la primera información del vector de movimiento o la segunda información del vector se determinan a partir del predictor del vector de movimiento seleccionado. La lista de candidatos del vector de movimiento del nivel de la unidad de predicción incluye al menos uno de los siguientes: candidatos contiguos espaciales y candidatos contiguos temporales. La lista de candidatos de predicción del movimiento del nivel de la unidad de subpredicción incluye al menos uno de los siguientes: predicción de vectores de movimiento afín, predicción del vector de movimiento

temporal alternativo (ATMVP), predicción del vector de movimiento espacial-temporal (STMVP), predicción del vector de movimiento plano, y derivación del vector de movimiento emparejado por patrón (PMVD). El procesador configurado además para derivar un índice de candidato de fusión en respuesta a determinar que la bandera de movimiento de la unidad de subpredicción está activa, donde el índice de candidato de fusión especifica el predictor del vector de movimiento a seleccionar.

[08] En otra realización de ejemplo, se discute un aparato para decodificar los datos de video. El aparato incluye un medio de memoria para almacenar una información de video codificada y recibida, y un medio de procesador. El medio de procesador configurado para analizar una bandera de movimiento de la unidad de subpredicción a partir de los datos de video codificados, en respuesta a determinar que la bandera de movimiento de la unidad de subpredicción está activa, derivar una lista de candidatos de predicción del movimiento del nivel de la unidad de subpredicción, en respuesta a determinar que la bandera de movimiento de la unidad de subpredicción no está activa, derivar una lista de candidatos de predicción del movimiento del nivel de la unidad de predicción, seleccionar un predictor del vector de movimiento de cualquiera de las listas, ya sea la de candidatos de predicción del movimiento del nivel de la unidad de subpredicción o de la lista de candidatos de predicción del movimiento del nivel de la unidad de predicción, y decodificar los datos de video codificados mediante el predictor del vector de movimiento seleccionado. Los datos de video codificados incluyen un bloque actual y donde la lista de candidatos de predicción del movimiento del nivel de la unidad de subpredicción y la lista de candidatos de predicción del movimiento del nivel de la unidad de predicción se derivan de bloques contiguos del bloque actual. Los bloques contiguos son contiguos espaciales del bloque actual en una imagen actual o contiguos temporales del bloque actual en una imagen codificada previamente. La lista de candidatos de predicción del movimiento del nivel de la unidad de subpredicción o la lista de candidatos de predicción del movimiento del nivel de la unidad de predicción se ordena al menos parcialmente con base en ocurrencias de predicción de movimiento de los bloques contiguos. Los píxeles en una unidad de predicción comparten una primera información del vector de movimiento y

los píxeles en una unidad de subpredicción comparten una segunda información del vector de movimiento, y donde la primera información del vector de movimiento o la segunda información del vector se determinan a partir del predictor del vector de movimiento seleccionado. La lista de candidatos del vector de movimiento del nivel de la unidad de predicción incluye al menos uno de los siguientes: candidatos contiguos espaciales y candidatos contiguos temporales. La lista de candidatos de predicción del movimiento del nivel de la unidad de subpredicción incluye al menos uno de los siguientes: predicción de vectores de movimiento afín, predicción del vector de movimiento temporal alternativo (ATMVP), predicción del vector de movimiento espacial-temporal (STMVP), predicción del vector de movimiento plano, y derivación del vector de movimiento emparejado por patrón (PMVD). El medio de procesador está configurado además para derivar un índice de candidato de fusión en respuesta a determinar que la bandera de movimiento de la unidad de subpredicción está activa, donde el índice de candidato de fusión especifica el predictor del vector de movimiento a seleccionar.

[09] En otra realización de ejemplo, un medio de almacenamiento no transitorio legible por computadora que tiene almacenadas instrucciones que, cuando se ejecutan, hacen que uno o más procesadores realicen un método. El método incluye recibir datos de video codificados, analizar una bandera de movimiento de la unidad de subpredicción a partir de los datos de video codificados, en respuesta a determinar que la bandera de movimiento de la unidad de subpredicción está activa, derivar una lista de candidatos de predicción del movimiento del nivel de la unidad de subpredicción, en respuesta a determinar que la bandera de movimiento de la unidad de subpredicción no está activa, derivar una lista de candidatos de predicción del movimiento del nivel de la unidad de predicción, seleccionar un predictor del vector de movimiento de cualquiera de la listas, ya sea la de candidatos de predicción del movimiento del nivel de la unidad de subpredicción o de la lista de candidatos de predicción del movimiento del nivel de la unidad de predicción, y decodificar los datos de video codificados mediante el predictor del vector de movimiento seleccionado. Los datos de video codificados incluyen un bloque actual y donde la lista de candidatos de

predicción del movimiento del nivel de la unidad de subpredicción y la lista de candidatos de predicción del movimiento del nivel de la unidad de predicción se derivan de bloques contiguos del bloque actual. Los bloques contiguos son contiguos espaciales del bloque actual en una imagen actual o contiguos temporales del bloque actual en una imagen codificada previamente. La lista de candidatos de predicción del movimiento del nivel de la unidad de subpredicción o la lista de candidatos de predicción del movimiento del nivel de la unidad de predicción se ordena al menos parcialmente con base en ocurrencias de predicción de movimiento de los bloques contiguos. Los píxeles en una unidad de predicción comparten una primera información del vector de movimiento y los píxeles en una unidad de subpredicción comparten una segunda información del vector de movimiento, y donde la primera información del vector de movimiento o la segunda información del vector se determinan a partir del predictor del vector de movimiento seleccionado. La lista de candidatos del vector de movimiento del nivel de la unidad de predicción incluye al menos uno de los siguientes: candidatos contiguos espaciales y candidatos contiguos temporales. La lista de candidatos de predicción del movimiento del nivel de la unidad de subpredicción incluye al menos uno de los siguientes: predicción de vectores de movimiento afín, predicción del vector de movimiento temporal alternativo (ATMVP), predicción del vector de movimiento espacial-temporal (STMVP), predicción del vector de movimiento plano, y derivación del vector de movimiento emparejado por patrón (PMVD). El método incluye derivar un índice de candidato de fusión en respuesta a determinar que la bandera de movimiento de la unidad de subpredicción está activa, donde el índice de candidato de fusión especifica el predictor del vector de movimiento a seleccionar.

[010] Los detalles de una o más ejemplos se exponen en las figuras adjuntas y la siguiente descripción. Otras características, objetivos y ventajas serán evidentes a partir de la descripción, figuras y reivindicaciones.

BREVE DESCRIPCIÓN DE LAS FIGURAS

[011] 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.

[012] Las FIGS. 2A y 2B son diagramas conceptuales que ilustran un ejemplo de estructura de árbol cuádruple-árbol binario (QTBT), y una unidad de árbol de codificación correspondiente (CTU).

[013] La FIG. 3 es un diagrama de flujo que ilustra una derivación del predictor del vector de movimiento.

[014] La FIG. 4 es un diagrama conceptual que ilustra candidatos del vector del movimiento contiguo espacial para un modo de fusión.

[015] La FIG. 5 es un diagrama conceptual que ilustra candidatos del vector de movimiento temporal.

[016] La FIG. 6 ilustra técnicas de selección de bloques del vector de movimiento candidato.

[017] La FIG. 7 es un diagrama conceptual que ilustra Predicción del Vector de Movimiento Espacial-Temporal (STMVP).

[018] La FIG. 8 es un diagrama conceptual que ilustra técnicas de emparejamiento bilateral.

[019] La FIG. 9 es un diagrama conceptual que ilustra técnicas de emparejamiento de plantilla.

[020] La FIG. 10 es un diagrama conceptual que ilustra la predicción de vectores de movimiento plano.

[021] La FIG. 11 es un diagrama de bloques que ilustra un codificador de video de ejemplo que puede realizar las técnicas de la presente divulgación.

[022] La FIG. 12 es un diagrama de bloques que ilustra un ejemplo del decodificador de video que puede realizar las técnicas de la presente divulgación.

DESCRIPCIÓN DETALLADA

[023] Esta divulgación se refiere a derivación de vectores de movimiento del lado del decodificador (DMVD). Las técnicas de derivación de vectores de movimiento del lado del decodificador descritas en esta divulgación pueden utilizarse en conjunción con

cualquiera de los códecs de video existentes, tal como HEVC (High Efficiency Video Coding), o pueden utilizarse como técnicas de codificación para cualquier estándar de codificación de video futuro, tal como H.266 Versatile Video Coding (VVC) y Essential Video Coding (EVC).

[024] 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 son generalmente dirigidas a la codificación (codificación y/o decodificación) de datos de video y, en particular, están relacionadas con técnicas discutidas en la presente. En general, los datos de video incluyen todo dato para el procesamiento de un video. Por lo tanto, los datos de video pueden incluir, video en bruto, video sin codificar, video codificado, video decodificado (por ejemplo, reconstituido), y metadatos de video, tal como datos de señalización.

[025] Como se muestra en la FIG. 1, el sistema 100 incluye un dispositivo fuente 102 que proporciona datos de video codificados a decodificar y mostrar por un dispositivo diana 116 en este ejemplo. En particular, el dispositivo fuente 102 proporciona los datos de video al dispositivo diana 116 a través de un medio legible por computadora 110. El dispositivo fuente 102 y el dispositivo diana 116 pueden comprender cualquiera de un amplio rango de dispositivos, que incluye computadoras de escritorio, dispositivos informáticos móviles, computadoras portátiles (por ejemplo, un portátil), computadoras tipo tableta, decodificadores, aparatos telefónicos, tal como los *smartphones*, televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de juegos de video, dispositivos de transmisión de videos en directo, o similares. En algunos casos, el dispositivo fuente 102 y el dispositivo diana 116 pueden estar equipados para la comunicación inalámbrica y, por lo tanto, pueden denominarse dispositivos de comunicación inalámbrica.

[026] En el ejemplo de la FIG. 1, el dispositivo fuente 102 incluye una fuente de video 104, memoria 106, codificador de video 200 e interfaz de salida 108. El dispositivo diana 116 incluye una interfaz de entrada 122, un decodificador de video 300, una memoria 120, y dispositivo de visualización 118. De acuerdo con la presente

divulgación, el codificador de video 200 de la fuente de video 102 y el decodificador de video 300 del dispositivo diana 116 pueden estar configurados para aplicar las técnicas para la derivación del vector del movimiento del lado del decodificador. Por lo tanto, el dispositivo fuente 102 representa un ejemplo de un dispositivo de codificación de video, mientras que el dispositivo diana 116 representa un ejemplo de un dispositivo de decodificación de video. En otros ejemplos, un dispositivo fuente y un dispositivo diana pueden incluir otros componentes o disposiciones. Por ejemplo, el dispositivo fuente 102 puede recibir datos de video desde una fuente de video externa, tal como una cámara externa. De manera similar, el dispositivo diana 116 puede interconectar con un dispositivo de visualización externo, en lugar de incluir un dispositivo de visualización integrado.

[027] El sistema 100 como se muestra en la FIG. 1 es meramente un ejemplo. En general, cualquier dispositivo de codificación y/o decodificación de video digital puede realizar técnicas para la derivación del vector de movimiento del lado del decodificador. El dispositivo fuente 102 y el dispositivo diana 116 son meramente ejemplos de tales dispositivos de codificación donde el dispositivo fuente 102 genera datos codificados de video para su transmisión al dispositivo diana 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, los dispositivos 102, 116 pueden operar de una manera sustancialmente simétrica de tal manera que cada uno de los dispositivos 102, 116 incluyen componentes de codificación y decodificación de video. Por lo tanto, el sistema 100 puede soportar transmisión de video unidireccional o bidireccional entre dispositivos de video 102, 116, por ejemplo, para transmisión de video en directo, reproducción de video, difusión de video, o telefonía de video.

[028] En general, la fuente de video 104 representa una fuente de datos de video (es decir, datos de video sin codificar, en bruto) y proporciona una serie secuencial de imágenes (también denominada "cuadros") de los datos de video al codificador de

video 200, que codifica los datos para las imágenes. La fuente de video 104 del dispositivo fuente 102 puede incluir un dispositivo de captura de video, tal como una cámara de video, un archivo de video que contiene video en bruto previamente capturado, y/o una interfaz de entrada de video para recibir el video de un proveedor de contenido de video. Como otra alternativa, la fuente de video 104 puede generar datos basados en gráficos por computadora como la fuente de video, o una combinación de video en directo, video archivado, y video generado por computadora. En cada caso, el codificador de video 200 codifica los datos de video capturados, precapturados, o generados por computadora. El codificador de video 200 puede reordenar las imágenes a partir del orden recibido (a menudo denominado "orden de presentación") 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 fuente 102 puede entonces dar salida a los datos de video codificados a través de la interfaz de salida 108 en un medio legible por computadora 110 para la recepción y/o recuperación mediante, por ejemplo, interfaz de entrada 122 del dispositivo diana 116.

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

para almacenar datos de video en bruto, decodificados, y/o codificados.

[030] 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 fuente 102 al dispositivo diana 116. En un ejemplo, el medio legible por computadora 110 representa un medio de comunicación para permitir al dispositivo fuente 102 transmitir datos de video codificados directamente a un dispositivo diana 116 en tiempo real, por ejemplo, a través de una red de frecuencia de radio o red basada en computadora. 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 modular la señal de transmisión recibida, de acuerdo con un estándar de comunicación, tal como un protocolo de comunicación inalámbrica. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrica o por cable, tal como un espectro de frecuencia de radio (RF) o una o más líneas de transmisión físicas. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área amplia, o una red global tal como Internet. El medio de comunicación puede incluir enrutadores, conmutadores, estaciones base, o cualquier otro equipo que puede ser útil para facilitar la comunicación desde el dispositivo fuente 102 al dispositivo diana 116.

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

[032] En algunos ejemplos, el dispositivo fuente 102 puede dar salida a datos de video codificados al servidor de archivos 114 u otro dispositivo de almacenamiento intermedio que puede almacenar el video codificado generado por el dispositivo

fuente 102. El dispositivo diana 116 puede acceder a los datos de video almacenados del servidor de archivos 114 a través de *streaming* o descarga. El servidor de archivos 114 puede ser cualquier tipo de servidor capaz de almacenar datos de video codificados y de transmitir los datos de video codificados al dispositivo diana 116. Los servidores de archivos 114 pueden representar un servidor web (por ejemplo, para un sitio web), un servidor de Protocolo de Transferencia de Archivos (FTP), un dispositivo de red de administración de contenidos, o un dispositivo de almacenamiento en red adjunta (NAS). El dispositivo diana 116 puede acceder a los datos de video codificados del servidor de archivos 114 a través de cualquier conexión de datos estándar, que incluye una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión wifi), una conexión por cable (por ejemplo, DSL, módem de cable, etc.), o una combinación que es adecuada para acceder a los datos de video codificados almacenados en un servidor de archivos 114. El servidor de archivos 114 y la interfaz de entrada 122 pueden estar configurados para operar de acuerdo con un protocolo de transmisión de *streaming*, un protocolo de transmisión de descarga, o una combinación de ellos.

[033] La interfaz de salida 108 y la interfaz de entrada 122 pueden representar transmisores/receptores inalámbricos, módems, componentes de red cableada (por ejemplo, tarjetas de Ethernet), componentes de comunicación inalámbrica que operan de acuerdo con cualquiera de una variedad de estándares IEEE 802.11, u otros componentes físicos. En ejemplos donde la interfaz de salida 108 y la interfaz de entrada 122 comprenden componentes inalámbricos, la interfaz de salida 108 y la interfaz de entrada 122 se pueden configurar para transferir datos, tal como datos de video codificados, de acuerdo con un estándar de comunicación celular, tal como 4G, 4G-LTE (Evolución a Largo Plazo), LTE Avanzada, 5G, o similares. En algunos ejemplos donde la interfaz de salida 108 comprende un transmisor inalámbrico, la interfaz de salida 108 y la interfaz de entrada 122 pueden estar configuradas para transferir datos, tal como datos de video codificados, de acuerdo con otros estándares inalámbricos, tal como una especificación IEEE 802. 11, una especificación IEEE 802.15 (por ejemplo, ZigBee ™), un estándar Bluetooth ™, o similares. En algunos ejemplos, el dispositivo

fuente 102 y/o el dispositivo diana 116 pueden incluir dispositivos de sistema en un chip respectivos (SoC). Por ejemplo, el dispositivo fuente 102 puede incluir un dispositivo SoC para realizar la funcionalidad atribuida al codificador de video 200 y/o a la interfaz de salida 108, y el dispositivo diana 116 puede incluir un dispositivo SoC para realizar la funcionalidad atribuida al decodificador de video 300 y/o a la interfaz de entrada 122.

[034] Las técnicas de la presente divulgación pueden aplicarse a la codificación de video en apoyo de cualquiera de una variedad de aplicaciones multimedia, tal como transmisiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de video en directo de Internet, tal como *streaming* adaptativo dinámico a través de HTTP (DASH), video digital que está codificado sobre un medio de almacenamiento de datos, decodificación de video digital almacenado en un medio de almacenamiento de datos, u otras aplicaciones.

[035] La interfaz de entrada 122 del dispositivo diana 116 recibe un flujo de bits de video codificado de medio legible por computadora 110 (por ejemplo, dispositivo de almacenamiento no transitorio 112, servidor de archivos 114, o similares). El medio legible por computadora de flujo de bits de video codificado 110 puede incluir información definida por el codificador de video 200, que también se utiliza por el decodificador de video 300, tal como los elementos de sintaxis que tienen valores que describen características y/o procesamiento de los bloques de video u otras unidades codificadas (por ejemplo, porciones, 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 tal como un tubo de rayos catódicos (CRT), una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodo luminoso orgánico (OLED), o cualquier tipo de dispositivo de visualización.

[036] 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 adecuadas,

u otro hardware y/o software, para manejar flujos multiplexados que incluyen audio y video en un flujo de datos común. Si es aplicable, las unidades MUX-DEMUX pueden ajustarse al protocolo multiplexor ITU H.223, u otros protocolos tal como el protocolo de datagrama de usuario (UDP).

[037] 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, tal como uno o más microprocesadores, procesadores de señal digital (DSP), circuitos integrados de aplicación específica (ASIC), matrices de puertas programables en campo (FPGA), lógica discreta, hardware, o cualquiera de sus combinaciones. Si las técnicas se implementan parcialmente en software, un dispositivo puede almacenar las instrucciones para el software en un medio adecuado de almacenamiento legible por computadora no transitorio y puede ejecutar las instrucciones en hardware utilizando uno o más procesadores para llevar a cabo las técnicas de esta divulgación. Cada uno del codificador de video 200 y decodificador de video 300 puede incluirse en uno o más codificadores o decodificadores, cualquiera de los cuales pueden estar integrados como parte de un codificador/decodificador combinado (CODEC) en un dispositivo respectivo. Un dispositivo que incluye un codificador de video 200 y/o decodificador de video 300 puede comprender un circuito integrado, un microprocesador, y/o un dispositivo de comunicación inalámbrica, tal como un teléfono celular.

[038] Los estándares de codificación de video incluyen ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 o ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual e ITU-T H.264 (también denominado ISO/IEC MPEG-4 AVC), incluyendo sus extensiones de Scalable Video Coding (SVC) y Multi-view Video Coding (MVC).

[039] Un estándar de codificación de video, a saber High Efficiency Video Coding (HEVC) o ITU-T H.265, que incluye su extensión de rango, extensión de múltiples vistas (MV-HEVC) y extensión escalable (SHVC), ha sido desarrollado por el Joint Collaboration Team on Video Coding (JCT-VC), así como Joint Collaboration Team on 3D Video Coding Extension Development (JCT-3V) del ITU-T Video Coding Experts Group (VCEG) e ISO/IEC Motion Picture Experts Group (MPEG).

[040] ITU-T VCEG (Q6/16) e ISO/IEC MPEG (JTC 1/SC 29/WG 11) están estudiando la posible necesidad de estandarización de la futura tecnología de codificación de video con una capacidad de compresión que supera a la del estándar de HEVC actual (incluyendo sus extensiones actuales y extensiones a corto plazo para la codificación de contenido de la pantalla y codificación de alto rango dinámico). Los grupos están trabajando juntos en esta actividad de exploración en un esfuerzo de colaboración conjunta conocida como el Joint Video Exploration Team (JVET) para evaluar diseños de tecnología de compresión propuestos por los expertos en esta área. El JVET se reunió por primera vez del 19 al 21 octubre de 2015. Y la última versión del software de referencia, es decir, Joint Exploration Model 7 (JEM 7) puede descargarse desde:
https://jvet.hhi.fraunhofer.de/svn/svn_HMJEMSoftware/tags/HM-16.6-JEM-57.0/
Una descripción del algoritmo de Joint Exploration Test Model 7 (JEM7) puede denominarse JVET-G1001.

[041] El codificador de video 200 y el decodificador de video 300 pueden operar de acuerdo con un estándar de codificación de video, tal como ITU-T H.265, también denominado High Efficiency Video Coding (HEVC) o sus extensiones, tal como extensiones de múltiples vistas y/o codificación de video escalable. Alternativamente, el codificador de video 200 y el decodificador de video 300 pueden funcionar de acuerdo con otros estándares registrados o de la industria, tal como el Joint Exploration Test Model (JEM) o ITU-T H.266, también denominado Versatile Video Coding (VVC). Un proyecto reciente del estándar VVC se describe en Bross, et al. "Versatile Video Coding (Draft 3)," Joint Video Experts Team (JVET) de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 13th Meeting: Marrakech, MA, 9-18 January 20190, JVET-M1001-v3 (en adelante "VVC Proyecto 4"). Las técnicas de la presente divulgación, sin embargo, no se limitan a ningún estándar de codificación particular.

Como se discutirá, el decodificador de video 300 puede estar configurado para realizar una o más técnicas de la presente divulgación. Por ejemplo, el decodificador de video 300 puede analizar una bandera de movimiento de la unidad de subpredicción a partir de datos de video codificados y recibidos, derivar una lista de candidatos de predicción del movimiento del nivel de la unidad de subpredicción si la

bandera de unidad de subpredicción de movimiento está activa, derivar una lista de candidatos de predicción del movimiento del nivel de la unidad de predicción si la bandera de movimiento de la unidad de subpredicción no está activa, y decodificar los datos de video codificados utilizando un predictor del vector de movimiento seleccionado.

[042] En general, el codificador de video 200 y el decodificador de video 300 pueden realizar la codificación basada en bloques de imágenes. El término "bloque" generalmente se refiere a una estructura que incluye datos a procesar (por ejemplo, codificados, decodificados, o utilizados de otro modo en el proceso de codificación y/o decodificación). Por ejemplo, un bloque puede incluir una matriz de dos dimensiones de muestras de datos de luminancia y/o crominancia. En general, el codificador de video 200 y el decodificador de video 300 pueden codificar los 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 decodificador de video 300 pueden codificar componentes de luminancia y de crominancia, donde los componentes de crominancia pueden incluir componentes de crominancia de tonalidad roja y tonalidad azul. En algunos ejemplos, el codificador de video 200 convierte los datos con formato RGB recibidos a una representación YUV antes de la codificación, y el decodificador de video 300 convierte la representación YUV al formato RGB. Alternativamente, las unidades de pre y posprocesamiento (no mostrado) pueden llevar a cabo estas conversiones.

[043] Esta divulgación puede referirse en general a la codificación (por ejemplo, codificación y decodificación) de imágenes para incluir el proceso de codificación o decodificación de datos de la imagen. Del mismo modo, 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 incluye generalmente una serie de valores para elementos de sintaxis representantes de decisiones de codificación (por ejemplo, modos de codificación) y la partición de las imágenes en bloques. Por lo tanto, las referencias a la codificación de una imagen o un bloque

generalmente deben comprenderse como valores de codificación para los elementos de sintaxis que forman la imagen o el bloque.

[044] HEVC define diversos bloques, incluidas las unidades de codificación (CU), unidades de predicción (PU) y unidades de transformada (TU). De acuerdo con HEVC, un codificador de video (tal como el codificador de video 200) particiona una unidad de árbol de codificación (CTU) en CU de acuerdo con una estructura de árbol cuádruple. Es decir, el codificador de video particiona CTU y CU en cuatro cuadrados iguales, que no se solapan, y cada nodo del árbol cuádruple tiene cero o cuatro nodos hijos. Los nodos sin nodos hijos pueden denominarse "nodos de hoja", y CU de tales nodos de hoja pueden incluir una o más unidades PU y/o una o más TU. El codificador de video puede particionar más PU y TU. Por ejemplo, en HEVC, un árbol cuádruple residual (RQT) representa la partición de TU. En HEVC, PU representan datos interpredicción, mientras que las TU representan datos residuales. CU que están intrapredichas incluyen información intrapredicción, tal como una indicación de intramodo.

[045] Como otro ejemplo, el codificador de video 200 y el decodificador de video 300 pueden estar configurados para funcionar de acuerdo con JEM o VVC. De acuerdo con JEM o VVC, un codificador de video (tal como el codificador de video 200) particiona una imagen en una pluralidad de unidades de árbol de codificación (CTU). El codificador de video 200 puede particionar una CTU de acuerdo con una estructura de árbol, tal como una estructura de árbol cuádruple-árbol binario (QTBT). La estructura QTBT elimina los conceptos de múltiples tipos de partición, tal como la separación entre CU, PU, y TU de HEVC. Una estructura QTBT de JEM incluye dos niveles: un primer nivel con particiones de acuerdo con la partición de árbol cuádruple, y un segundo nivel con particiones de acuerdo con la partición de árbol binario. Un nodo de raíz de la estructura QTBT corresponde a una CTU. Los nodos de hoja de los árboles binarios corresponden a las unidades de codificación (CU).

[046] En una estructura de partición MTT, los bloques se pueden particionar utilizando una partición de árbol cuádruple (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 un bloque que se divide en tres subbloques. En algunos ejemplos, una partición de árbol triple se divide en un bloque en tres subbloques sin dividir el bloque original a través del centro. Los tipos de partición en MTT (por ejemplo, QT, BT, y TT) pueden ser simétricos o asimétricos.

[047] 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 decodificador 300 de video pueden utilizar dos o más estructuras QTBT o MTT , tal como una estructura QTBT/MTT para el componente de luminancia y otra estructura QTBT o MTT para ambos componentes de crominancia (o dos estructuras QTBT/MTT para los respectivos componentes de crominancia).

[048] El codificador de video 200 y el decodificador de video 300 pueden estar configurados para usar la partición de árbol cuádruple por HEVC, la partición de QTBT o una partición de MTT u otras estructuras de partición. Con un propósito explicativo, la descripción de las técnicas de esta divulgación se presenta con respecto a partición de QTBT. Sin embargo, se debe comprender que las técnicas de la presente divulgación también se pueden aplicar a codificadores de video configurados para utilizar la partición de árbol cuádruple, u otros tipos de partición.

[049] Esta divulgación puede utilizar “NxN” y “N por N” de manera intercambiable para referirse a las dimensiones de muestra de un bloque (tal como una CU u otro bloque de video) en términos de las dimensiones verticales y horizontales, por ejemplo, muestras de 16x16 o muestras de 16 por 16. En general, una CU de 16x16 tiene 16 muestras en una dirección vertical (Y = 16) y 16 muestras en una dirección horizontal (x = 16). Del mismo modo, una CU de NxN generalmente tiene N muestras en una dirección vertical y N muestras en una dirección horizontal, donde N representa un valor entero no negativo. Las muestras en una CU pueden estar dispuestas en filas y columnas. Por otra parte, las CU no necesariamente tienen por qué tener el mismo número de muestras en la dirección horizontal que en la dirección vertical. Por ejemplo, las CU pueden comprender muestras NxM, donde M no es necesariamente igual a N.

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

[051] Para predecir una CU, el codificador de video 200 puede generalmente formar un bloque de predicción para la CU a través de la interpredicción o intrapredicción. La interpredicción generalmente se refiere a la predicción de la CU a partir de datos de una imagen codificada previamente, 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 interpredicción, el codificador de video 200 puede generar el bloque de predicción utilizando 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 se acerque a 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 usando una suma de la diferencia absoluta (SAD), suma de diferencias cuadradas (SSD), media de diferencia absoluta (MAD), media de diferencias al cuadrado (MSD), u otros cálculos de dicha diferencia para determinar si un bloque de referencia coincide estrechamente con la CU actual. En algunos ejemplos, el codificador de video 200 puede predecir la CU actual utilizando la predicción unidireccional o predicción bidireccional.

[052] Algunos ejemplos de JEM y VVC también proporcionan 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 el movimiento no traslacional, tal como el acercamiento o alejamiento, rotación, movimiento de perspectiva, u otros tipos de movimiento irregulares.

[053] Para llevar a cabo intrapredicción, el codificador de video 200 puede seleccionar un modo de intrapredicción para generar el bloque de predicción. Algunos ejemplos de JEM y VVC proporcionan sesenta y siete modos de intrapredicción,

incluyendo varios modos de dirección, así como el modo plano y el modo DC. En general, el codificador de video 200 selecciona un modo de intrapredicción que describe muestras contiguas para un bloque actual (por ejemplo, un bloque de una CU) del que predecir muestras del bloque actual. Dichas muestras pueden estar generalmente por 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 codificar CTU y CU en orden de escaneo rasterizado (de izquierda a derecha, de arriba a abajo).

[054] 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 los datos que representan cuál de los varios modos de interpredicción disponible se utiliza, así como la información de movimiento para el modo correspondiente. Para interpredicción unidireccional o bidireccional, por ejemplo, el codificador de video 200 puede codificar vectores de movimiento utilizando la predicción de vectores de movimiento avanzada (AMVP) o el modo de fusión. El codificador de video 200 puede utilizar modos similares para codificar vectores de movimiento para el modo de compensación de movimiento afín.

[055] Después de la predicción, tal como la intrapredicción o interpredicción de un bloque, el codificador de video 200 puede calcular los datos residuales para el bloque. Los datos residuales, tal como un bloque residual, representan la muestra por las diferencias de muestra entre el bloque y un bloque de predicción para el bloque, formada usando 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 la transformada en lugar del dominio de la muestra. Por ejemplo, el codificador de video 200 puede aplicar una transformada de coseno discreta (DCT), una transformada entera, una transformada de ondícula, o una transformada conceptualmente similar a los datos de video residuales. Adicionalmente, el codificador de video 200 puede aplicar una transformada secundaria después de la primera transformada, tal como una transformada secundaria no separable dependiente del modo (MDNSST), una transformada

dependiente de la señal, una transformada Karhunen-Loeve (KLT), o similares. El codificador de video 200 produce coeficientes de transformada después de la aplicación de las una o más transformadas.

[056] Como se señaló anteriormente, después de cualquier transformada para producir coeficientes de transformada, el codificador de video 200 puede realizar la cuantificación de los coeficientes de transformada. La cuantificación se refiere generalmente a un proceso donde se cuantifican coeficientes de transformada para reducir posiblemente la cantidad de datos utilizados para representar los coeficientes, proporcionando una compresión adicional. Por 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 hacia abajo a un valor de *m*-bits durante la cuantificación, donde *n* es mayor que *m*. En algunos ejemplos, para llevar a cabo la cuantificación, el codificador de video 200 puede llevar a cabo un desplazamiento a la derecha bit a bit del valor a cuantificar.

[057] Tras la cuantificación, el codificador de video 200 puede barrer los coeficientes de transformada, producir un vector unidimensional de la matriz de dos dimensiones, incluyendo los coeficientes de transformada cuantificados. El barrido puede estar diseñado para colocar coeficientes de mayor energía (y por lo tanto menor frecuencia) en la parte delantera del vector y para colocar coeficientes de menor (y por lo tanto mayor frecuencia) de transformada en la parte posterior del vector. En algunos ejemplos, el codificador de video 200 puede utilizar un orden de barrido predefinido para barrer los coeficientes de transformada cuantificados para producir un vector serializado y, a continuación, codificar por entropía los coeficientes de transformada cuantificados del vector. En otros ejemplos, el codificador de video 200 puede realizar un barrido adaptativo. Después de barrer los coeficientes de transformada cuantificados para formar un vector unidimensional, el codificador de video 200 puede codificar por entropía el vector unidimensional, por ejemplo, de acuerdo con codificación aritmética binaria de contexto adaptativo (CABAC). El codificador de video 200 también puede codificar por entropía elementos de sintaxis asociados con los

datos de video codificados para su uso por el decodificador de video 300 al decodificar los datos de video.

[058] Para llevar a cabo CABAC, el codificador de video 200 puede asignar un contexto dentro de un modelo de contexto a un símbolo a transmitir. El contexto puede referir a, por ejemplo, si los valores contiguos del símbolo son cero o no. La determinación de probabilidad se puede basar en un contexto asignado al símbolo.

[059] El codificador de video 200 puede generar datos de sintaxis adicionales, tal como datos de sintaxis basados en bloques, datos de sintaxis basados en imagen, y datos de sintaxis basados en secuencia, al decodificador de video 300, por ejemplo, en una cabecera de imagen, una cabecera de bloque, una cabecera de segmento, u otros datos de sintaxis, tal como un conjunto de parámetros de secuencia (SPS), conjunto de parámetros de imagen (PPS), o conjunto de parámetros de video (VPS). El decodificador de video 300 puede asimismo decodificar dichos datos de sintaxis para determinar cómo decodificar los datos de video correspondientes.

[060] 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 la predicción y/o información residual para los bloques. En última instancia, el decodificador de video 300 puede recibir el flujo de bits y decodificar los datos de video codificados.

[061] 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 los valores para elementos de sintaxis del flujo de bits utilizando CABAC de una manera sustancialmente similar, aunque recíproca, al proceso de codificación CABAC del codificador de video 200. Los elementos de sintaxis pueden definir la partición de información de una imagen en CTU, y la partición de cada CTU de acuerdo con una estructura de partición correspondiente, tal como una estructura QTBT, para definir CU de la CTU. Los elementos de sintaxis pueden definir aún más la predicción y la información residual para los bloques (por ejemplo, las CU) de datos de video.

[062] La información residual puede representarse, por ejemplo, por coeficientes de

transformada cuantificados. El decodificador de video 300 puede cuantificar en forma inversa y transformar en forma inversa los coeficientes de transformada cuantificados de un bloque para reproducir un bloque residual para el bloque. El decodificador de video 300 utiliza un modo de predicción señalizado (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 (sobre una base de muestra por muestra) para reproducir el bloque original. El decodificador de video 300 puede realizar el procesamiento adicional, tal como realizar un proceso de desbloqueo para reducir los artefactos visuales a lo largo de los límites del bloque.

[063] Esta divulgación puede referirse en general a la "señalización" de cierta información, tal como los elementos de sintaxis. El término "señalización" puede referirse en general a la comunicación de los valores de los elementos de sintaxis y/o de otros datos utilizados para decodificar datos de video codificados. Es decir, el codificador de video 200 puede ser una señal para los valores de los elementos de sintaxis en el flujo de bits. En general, la señalización se refiere a la generación de un valor en el flujo de bits. Como se señaló anteriormente, el dispositivo fuente 102 puede transportar el flujo de bits al dispositivo diana 116 sustancialmente en tiempo real, o no en tiempo real, tal como podría ocurrir durante el almacenamiento de elementos de sintaxis en el dispositivo de almacenamiento 112 para su posterior recuperación por el dispositivo diana 116.

[064] Las FIGS. 2A y 2B son diagramas conceptuales que ilustran un ejemplo de estructura de árbol cuádruple-árbol binario (QTBT) 130, y una unidad de árbol codificación correspondiente (CTU) 132. Las líneas continuas representan la división de árbol cuádruple, y las líneas de puntos indican la división de árbol binario. En cada división (es decir, no de hoja) de nodo del árbol binario se señaliza una bandera para indicar qué tipo de división se utiliza (es decir, horizontal o vertical), donde 0 indica división horizontal y 1 indica la división vertical en este ejemplo. Para la división de árbol cuádruple, no hay necesidad de indicar el tipo de división, dado que los nodos de

árbol cuádruple dividen un bloque horizontal y verticalmente en 4 subbloques con igual tamaño. En consecuencia, el codificador de video 200 puede codificar, y el decodificador de video 300 puede decodificar, elementos de sintaxis (tal como la información de división) para un nivel de árbol región de estructura QTBT 130 (es decir, las líneas continuas) y los elementos de sintaxis (tal como la información de división) para un nivel de árbol de predicción de la estructura QTBT 130 (es decir, las líneas de puntos). El codificador de video 200 puede codificar, y el decodificador de video 300 puede decodificar datos de video, tal como los datos de predicción y transformada, por CU representadas por nodos de hoja terminales de estructura QTBT 130.

[065] En general, la CTU 132 de la FIG. 2B puede estar asociada con parámetros que definen tamaños de bloques correspondientes a los nodos de la estructura QTBT 130 en los primeros y segundos niveles. Estos parámetros pueden incluir un tamaño CTU (que representa un tamaño de CTU 132 en las muestras), un tamaño mínimo de árbol cuádruple (MinQTSize, que representa un tamaño mínimo permitido de nodo de hoja de árbol cuádruple), un tamaño máximo de árbol binario (MaxBTSize, que representa un tamaño máximo permitido de nodo d raíz de árbol binario), una profundidad máxima de árbol binario (MaxBTDepth, que representa una profundidad máxima permitida de árbol binario), y un tamaño mínimo de árbol binario (MinBTSize, que representa el tamaño mínimo permitido de nodo de hoja de árbol binario).

[066] El nodo de 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 estar dividido de acuerdo con la partición de árbol cuádruple. Es decir, los nodos del primer nivel son nodos de hoja (que no tienen nodos hijos) o tienen cuatro nodos hijos. El ejemplo de la estructura QTBT 130 representa tales nodos como incluyendo el nodo padre y nodos secundarios que tiene líneas sólidas para ramas. Si los nodos del primer nivel no son más grandes que el tamaño máximo permitido del nodo de raíz del árbol binario (MaxBTSize), se pueden particionar adicionalmente en respectivos árboles binarios. La separación del árbol binario de un nodo puede repetirse hasta que los nodos resultantes de la separación alcanzan el tamaño mínimo

permitido de nodo de hoja de árbol binario (MinBTSize) o la profundidad máxima permitida de árbol binario (MaxBTDepth). El ejemplo de la estructura QTBT 130 representa tales nodos con líneas de puntos para las ramas. El nodo de hoja del árbol binario se denominada unidad de codificación (CU), que se utiliza para la predicción (por ejemplo, predicción intraimagen o interimagen) y transformada, sin ninguna partición adicional. Como se discutió anteriormente, las CU también puede denominarse “bloques de video” o “bloques”.

[067] En un ejemplo de la estructura de partición QTBT, el tamaño de CTU se establece como 128x128 (muestras de luminancia y dos muestras de crominancia 64x64 correspondientes), MinQTSize se establece como 16x16, MaxBTSize se establece como 64x64, MinBTSize (tanto para ancho y altura) se establece en 4, y MaxBTDepth se establece en 4. La partición de árbol cuádruple se aplica a la CTU primero para generar nodos de hoja de árbol cuádruple. Los nodos de hoja de árbol cuádruple pueden tener un tamaño de 16x16 (es decir, MinQTSize) a 128x128 (es decir, el tamaño de CTU). Si el nodo de hoja de árbol cuádruple es de 128x128, no se divide adicionalmente por el árbol binario, ya que el tamaño excede el MaxBTSize (es decir, 64x64, en este ejemplo). De lo contrario, el nodo de hoja de árbol cuádruple se divide en forma adicional por el árbol binario. Por lo tanto, el nodo de hoja de árbol cuádruple es también el nodo de raíz para el 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 división horizontal adicional. Cuando el nodo del árbol binario tiene ancho igual a MinBTSize (4, en este ejemplo), que implica no se permite la división horizontal adicional. Del mismo modo, un nodo de árbol binario que tiene una altura igual a MinBTSize implica que no se permite la división vertical adicional para ese nodo del árbol binario. Como se señaló anteriormente, los nodos de hoja del árbol binario se denominan CU, y se procesan adicionalmente de acuerdo con la predicción y transformada sin más partición.

[068] En esta sección, se discuten estándares de codificación de video, en especial técnicas relacionadas a la predicción del vector de movimiento de los estándares anteriores.

[069] Los estándares de codificación de video incluyen ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 o ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual e ITU-T H.264 (también denominado ISO/IEC MPEG-4 AVC), que incluye sus extensiones Scalable Video Coding (SVC) y Multi-View Video Coding (MVC). El último proyecto conjunto de MVC se describe en "Advanced video coding for generic audiovisual services," ITU-T Recommendation H.264, Mar. 2010.

[070] Además, hay un estándar de codificación de video recientemente desarrollado, a saber, High Efficiency Video Coding (HEVC), desarrollado por el Joint Collaboration Team on Video Coding (JCT-VC) of ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Motion Picture Experts Group (MPEG). Un proyecto reciente de HEVC está disponible en http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v34.zip.

[071] Información de movimiento

[072] Para cada bloque, puede estar disponible un conjunto de información de movimiento. El conjunto de información de movimiento contiene información de movimiento para las direcciones de predicción hacia delante y hacia atrás. En la presente, las direcciones de predicción hacia delante y hacia atrás son dos direcciones de predicción correspondientes a la lista de imágenes de referencia 0 (RefPicList0) y la lista de imágenes de referencia 1 (RefPicList1) de una imagen o segmento actual. Los términos "hacia adelante" y "hacia atrás" no necesariamente tienen un significado geométrico. En cambio, se utilizan para distinguir en qué lista de imágenes de referencia se basa un vector de movimiento. La predicción hacia delante significa la predicción formada con base en la lista de referencia 0, mientras que la predicción hacia atrás significa la predicción formada con base en la lista de referencia 1. En el caso en que se utilizan la lista de referencia 0 y la lista de referencia 1 para formar una predicción para un bloque dado, se denomina predicción bidireccional.

[073] Para una imagen o un segmento dados, si sólo se utiliza una lista de imágenes de referencia, cada bloque dentro de la imagen o segmento se predijo hacia adelante. Si se utilizan ambas listas de imágenes de referencia para una imagen o segmento dados, un bloque dentro de la imagen o segmento puede ser previsto hacia adelante, o

previsto hacia atrás, o previsto bidireccionalmente.

[074] Para cada dirección de predicción, la información de movimiento contiene un índice de referencia y un vector de movimiento. Un índice de referencia se utiliza 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 horizontal y uno vertical, donde cada uno indica un valor de desplazamiento a lo largo de la dirección horizontal y vertical respectivamente. En algunas descripciones, por simplicidad, el término "vector de movimiento" se puede utilizar de forma intercambiable con la información de movimiento, para indicar el vector de movimiento y su índice de referencia asociado.

[075] POC

[076] El conteo de orden de imágenes (POC) se utiliza en los estándares de codificación de video para identificar un orden de visualización de una imagen. Aunque hay casos en que dos imágenes dentro de una secuencia de video codificada pueden tener el mismo valor POC, por lo general no ocurre dentro de una secuencia de video codificada. Cuando múltiples secuencias de video codificados están presentes en un flujo de bits, las imágenes con un mismo valor de POC pueden estar más cerca una de la otra en términos de orden de decodificación.

[077] Los valores POC de imágenes se utilizan típicamente para la construcción de la lista de imágenes de referencia, la derivación de la imagen de referencia fijada como en HEVC y la escalada de vectores de movimiento.

[078] Unidad de predicción

[079] Una unidad de predicción (PU) se refiere a una unidad básica de muestras que comparten la misma información de predicción. En la PU de interpredicción, la información de predicción puede ser el mismo conjunto de información de movimiento o el mismo método para derivar la información de movimiento para la PU. En un ejemplo, la PU puede ser igual que en la estructura de bloque HEVC, y también puede ser el bloque básico en otras estructuras de bloques de partición, partición de árbol cuádruple más árbol binario, partición múltiples tipos de árbol, entre otros.

[080] Una sub-PU es un subbloque en una PU, donde las muestras comparten el

mismo conjunto de información de movimiento. En el movimiento de nivel sub-PU, las muestras en una PU comparten el mismo método/modelo para la derivación de información de movimiento, pero una sub-PU puede tener su propio conjunto de información de movimiento y puede ser diferente de otras sub-PU en la misma PU.

[081] La FIG. 3 ilustra un ejemplo de diagrama de flujo de derivación de predictor de vectores de movimiento, de acuerdo con lo discutido en la presente. Se derivan dos listas de candidatos de predictor del vector de movimiento de la información de los bloques contiguos en el dominio tridimensional. El decodificador selecciona adaptativamente una lista de acuerdo con un elemento de sintaxis decodificado *is_sub_pu_motion_flag* del flujo de bits.

[082] En un ejemplo, la lista sub_pu solamente puede existir si el bloque actual es mayor que un valor predefinido. En otro ejemplo, un bloque puede contener solamente una sub-PU, por ejemplo, bloque de 4x4, entonces la bandera es para indicar la generación de información de movimiento se basa en el método de sub-PU o no. Las dos listas pueden incluir un número diferente de candidatos de predicción del movimiento, que pueden estar predefinidos o señalizados en SPS, PPS o cabecera de segmento. Los bloques contiguos pueden ser los contiguos espaciales en la imagen actual o los contiguos temporales en las imágenes codificadas anteriores.

[083] Como se ilustra en la FIG. 3, se discute el proceso de derivación de las dos listas. En 300, el decodificador determina si la bandera está activa o no. Si el decodificador determina que la bandera no está activa, el decodificador procede a 302. Si la bandera está activa, el decodificador procede a 306.

[084] En 302, se deriva una primera lista de candidatos de predicción del movimiento de nivel de PU. En 306, se deriva una segunda lista de candidatos de predicción del movimiento de nivel de sub-PU. El candidato de predicción del movimiento de nivel de PU significa que todos los píxeles de la misma PU comparten el mismo conjunto de información de movimiento. El candidato de predicción del movimiento de nivel de sub-PU significa que todos los píxeles de la misma sub-PU comparten el mismo conjunto de información de movimiento, pero diferentes sub-PU en una PU pueden tener diferentes conjuntos de información de movimiento. Un conjunto de

información de movimiento puede incluir dirección interpredicción, índice de imagen de referencia o índices si se utilizan múltiples referencias, vector de movimiento o vectores de movimiento si se utilizan múltiples referencias.

[085] Un ejemplo de una lista de candidatos del vector de movimiento del nivel de PU es la lista de candidatos de fusión HEVC. Los ejemplos de predicción del movimiento del nivel de sub-PU incluyen, pero sin limitación, la Predicción de Vectores de Movimiento Afín (Afín), Predicción del Vector de Movimiento Temporal Alternativo (ATMVP), Predicción del Vector de Movimiento Espacial-Temporal (STMVP), Predicción del Vector de Movimiento Plano, y Derivación del Vector de Movimiento Emparejado por Patrón (PMVD), entre otros. Una tabla de sintaxis de ejemplo se muestra en la Tabla 1 a continuación. El elemento de sintaxis sub_pu_motion_idx puede utilizarse para indicar el candidato seleccionado en la lista de candidatos de predicción del movimiento del nivel de sub-PU, y el elemento de sintaxis pu_motion_idx se utiliza para indicar al candidato seleccionado en la lista de candidatos del movimiento del nivel de predicción de PU.

Tabla 1

	Descriptor
is_sub_pu_motion_flag	ae(v)
if *is_sub_pu_motion_flag* {	
sub_pu_motion_idx	ae(v)
}	
else {	
pu_motion_idx	ae(v)
}	

[086] En otro método, los candidatos del vector del movimiento del nivel de PU se pueden dividir en dos grupos. *is_sub_pu_motion_flag* se señaliza cuando el decodificador recibe la sintaxis que indica que el candidato seleccionado no está en el primer grupo de candidatos de información del movimiento del nivel de la PU. *is_sub_pu_motion_flag* puede señalizarse de forma implícita en el índice de fusión del

nivel de la PU como uno de los índices. Entonces el sub-índice de fusión del nivel de la PU se señaliza en forma adicional si *is_sub_pu_motion_flag* es verdadero. En un ejemplo, el orden de inserción de la lista de candidatos de predicción del movimiento del nivel de PU es A→B→S→C→D→E. A, B, C, D, E denotan los candidatos de predicción del movimiento del nivel de PU, y S es un indicador de sub_pu_motion_flag. Si se selecciona S, entonces sub_pu_motion_flag se infiere como verdadero. Un ejemplo de A puede ser el candidato de fusión espacial inferior izquierdo, y un ejemplo de B puede ser el candidato de fusión espacial superior derecho. Cabe destacar que también se puede aplicar otro orden de inserción.

[087] Cabe destacar que los candidatos de predicción del movimiento del nivel de sub-PU se agrupan en todos los métodos descritos anteriormente.

[088] Señalización de la selección de candidatos de predicción del vector de movimiento

[089] Un elemento de sintaxis *is_sub_pu_motion_flag* se utiliza en primer lugar para indicar si el candidato de predicción del movimiento es de nivel de sub-PU. El is_sub_pu_motion_flag puede binarizarse utilizando un bin (0/1) y codificarse por contexto con base en el codificador aritmético binario. El contexto puede ser dependiente del tamaño/área de la PU o la profundidad de la PU en el árbol de partición del bloque. Una PU más grande puede tender a tener más frecuencia en la selección de la predicción del vector del movimiento del nivel de sub-PU que una PU más pequeña. El contexto también puede depender de sub_pu_motion_flag a partir de bloques contiguos espacio/temporales. La posibilidad de que la PU actual utilice el movimiento de sub-PU es mayor si los bloques contiguos tienen movimiento de sub-PU.

[090] Si is_sub_pu_motion_flag es verdadero ("1"), un elemento de sintaxis *sub_pu_motion_idx* se utiliza para indicar el método para derivar el candidato de predicción del movimiento de sub-PU. El número total de métodos, es decir, número total de candidatos de predicción del movimiento del nivel de sub-PU, *num_sub_pu_motion* se puede señalizar en sintaxis de alto nivel. El sub_pu_motion_idx puede binarizarse utilizando el código unario truncado

dependiendo de num_sub_pu_motion. Sin embargo, también se puede aplicar otro método de binarización.

[091] Si is_sub_pu_motion_flag es falso (“0”), un elemento de sintaxis nor_pu_motion_idx se utiliza para indicar el método para derivar el candidato de predicción del movimiento del nivel de PU. El número total de candidatos de predicción del vector de movimiento del nivel de PU, *num_nor_pu_motion* se puede señalizar en la sintaxis de alto nivel. nor_pu_motion_idx puede binarizarse utilizando el código unario truncado dependiendo de num_nor_pu_motion. Sin embargo, también se puede aplicar otro método de binarización.

[092] Derivación de candidatos de predicción del movimiento del nivel de PU

[093] El candidato de predicción del movimiento del nivel de PU se puede derivar de los bloques espaciales o temporales codificados contiguos de manera similar en HEVC. En el modo de fusión HEVC.

[094] En HEVC, la lista de candidatos MV contiene hasta 5 candidatos para el modo de fusión y sólo 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 las dos listas de imágenes de referencia (lista 0 y lista 1) y los índices de referencia. Si un candidato de fusión se identifica por un índice de fusión, se determinan las imágenes de referencia utilizadas para la predicción de los bloques actuales, así como los vectores de movimiento asociados.

[095] Con base en lo anterior, se seleccionan uno o más predictores del vector de movimiento con base en un índice de candidato decodificado.

[096] En el modo de fusión HEVC, se pueden derivar hasta cuatro candidatos MV espaciales 402, 404, 406, y 408 de un bloque 400 con los órdenes mostrados en la FIG. 4. El orden es el siguiente: izquierdo (0, A1), superior (1, B1), arriba a la derecha (2, B0), abajo de la izquierda (3, A0), y arriba a la izquierda (4, B2).

[097] Se discuten a continuación los candidatos contiguos temporales en HEVC. Un candidato del predictor del vector de movimiento temporal (TMVP), si está activado y disponible, se añade a la lista de candidatos MV después de los candidatos del vector del movimiento espacial. La ubicación del bloque principal para la derivación del

candidato TMVP es el bloque abajo a la derecha fuera de la PU de ubicación compartida como se muestra en la FIG. 5 como un bloque “T” 500, para compensar el sesgo a los bloques superiores e izquierdos usados para generar candidatos contiguos espaciales. Sin embargo, si ese bloque 502 está situado fuera de la fila de CTB actual o la información de movimiento no está disponible, el bloque se sustituye con un bloque central 504 de la PU. El vector de movimiento para el candidato TMVP se deriva de la PU cosituada de la imagen cosituada, que se indica en el nivel de segmento. El vector de movimiento para la PU cosituada se denominada MV de ubicación compartida.

[098] Derivación de candidatos de predicción del movimiento del nivel de sub-PU

[099] Los candidatos de predicción del movimiento del nivel de sub-PU pueden incluir, pero se no limitan a, la Predicción de Movimiento Afín, Predicción del Vector de Movimiento Temporal Alternativo (ATMVP), Predicción del Vector de Movimiento Espacial-Temporal (STMVP), Predicción del Vector de Movimiento Plano, y Derivación del Vector de Movimiento Emparejado por Patrón (PMVD), et al. A continuación, se ilustran ejemplos de esta predicción del movimiento del nivel de sub-PU. Sin embargo, también se pueden añadir algunas variaciones u otra predicción de movimiento del nivel de sub-PU.

[0100] Predicción de movimiento afín

[0101] En el método de predicción del movimiento afín de cuatro parámetros, el campo del vector de movimiento de un bloque se describe por la ecuación (1):

$$\begin{cases} v_x = \frac{v_{1x}-v_{0x}}{w}x - \frac{v_{1y}-v_{0y}}{h}y + v_{0x} \\ v_y = \frac{v_{1y}-v_{0y}}{w}x - \frac{v_{1x}-v_{0x}}{h}y + v_{0y} \end{cases} \quad (1)$$

[0102] donde (w, h) es el tamaño del bloque, y (x, y) es la coordenada. (v_{0x}, v_{0y}) es el vector de movimiento del punto de control de la esquina superior izquierda, y (v_{1x}, v_{1y}) es el vector de movimiento del punto de control de la esquina superior derecha.

[0103] En el método de predicción del movimiento afín de seis parámetros, el campo de movimiento de un bloque se describe por la ecuación (2):

$$\begin{cases} v_x = \frac{v_{1x}-v_{0x}}{w}x - \frac{v_{2x}-v_{0x}}{h}y + v_{0x} \\ v_y = \frac{v_{1y}-v_{0y}}{w}x - \frac{v_{2y}-v_{0y}}{h}y + v_{0y} \end{cases} \quad (2)$$

[0104] donde, además, (v_{2x}, v_{2y}) es el vector de movimiento del punto de control de la esquina inferior izquierda.

[0105] En predicción del movimiento afín con base en sub-PU, el MV de una sub-PU se puede derivar mediante el cálculo del MV en el centro de la sub-PU. Alternativamente, podemos reducirse a (w, h) y (x, y) de acuerdo con el tamaño de sub-PU.

[0106] En un método, (v_{ix}, v_{iy}) se puede derivar de un bloque contiguo que está codificado previamente por el movimiento afín, teniendo en cuenta que los bloques actuales comparten el mismo modelo de movimiento afín con el bloque contiguo codificado anterior.

[0107] En otro método, (v_{ix}, v_{iy}) se puede derivar por el vector de movimiento en un bloque codificado contiguo. Por ejemplo, como se muestra en la FIG. 6, (v_{0x}, v_{0y}) se puede derivar del vector de movimiento en el bloque A 600, B 602, o C 604, (v_{1x}, v_{1y}) se puede derivar de vector de movimiento en el bloque C 606 o D 608, (v_{2x}, v_{2y}) se puede derivar del bloque E 610 o F 612.

[0108] Predicción del Vector de Movimiento Temporal Alternativo

[0109] En el método de predicción del vector de movimiento temporal alternativo (ATMVP) (o a menudo denominado método de predicción del vector de movimiento temporal avanzado), la predicción del vector de movimiento temporal (TMVP) se modifica al ir a buscar múltiples conjuntos de información de movimiento (incluyendo vectores de movimiento e índices de referencia) de las sub-PU de la PU actual.

[0110] Predicción del Vector de Movimiento Espacio-Temporal

[0111] En el método de predicción del vector de movimiento espacio-temporal, los vectores de movimiento de sub-PU se derivan de forma recursiva, siguiendo el orden de barrido de la trama. La FIG. 7 ilustra el concepto de STMVP. Considérese una PU 8x8 que contiene cuatro sub-UP 4x4 A, B, C, y D. Los bloques de 4x4 contiguos en la trama

actual son etiquetados como a, b, c, y d. La derivación del movimiento de sub-CU A inicia mediante la identificación de sus dos vecinos espaciales. El primer vecino es el bloque de N x N sobre sub-CU A (bloque C). Si este bloque c no está disponible o está intracodificado los otros bloques N×N sobre sub-CU A se verifican (de izquierda a derecha, comenzando en el bloque c). El segundo vecino es un bloque a la izquierda de la sub-CU A (bloque B). Si el bloque B no está disponible o está intracodificado otros bloques a la izquierda del sub-CU A se verifican (de arriba a abajo, comenzando en el bloque b). La información de movimiento obtenida de los bloques contiguos para cada lista se escala para la primera trama de referencia para una lista dada. Después, el predictor del vector del movimiento temporal (TMVP) del subbloque A se deriva siguiendo el mismo procedimiento de derivación de TMVP como se especifica en HEVC. La información de movimiento del bloque adjunto en la posición D es traída y reducida en consecuencia. Finalmente, después de la recuperación y ampliación de la información de movimiento, todos los vectores de movimiento disponibles (hasta 3) se promedian por separado para cada lista de referencia. El vector de movimiento promediado se asigna como el vector de movimiento de la sub-CU actual.

[0112] Derivación del Vector de Movimiento Emparejado por Patrón

[0113] El método de derivación del vector de movimiento emparejado por patrón (PMMVD) se basa en las técnicas de conversión de velocidad de cuadro. El proceso de derivación del movimiento tiene dos pasos. En primer lugar, se realiza una búsqueda de movimiento del nivel de PU, seguida a continuación por un refinamiento del movimiento del nivel de sub-PU. En el nivel de PU, un vector de movimiento inicial se deriva para toda la PU con base en el emparejamiento bilateral o emparejamiento de plantilla. En primer lugar, se genera una lista de candidatos de MV y el candidato que lleva al mínimo costo de emparejamiento se selecciona como el punto de partida para un mayor refinamiento del nivel de PU. A continuación, se lleva a cabo una búsqueda local basada en el emparejamiento bilateral o emparejamiento de plantilla alrededor del punto de partida y los resultados de MV en el mínimo costo de emparejamiento se toman como el MV para toda la CU. Posteriormente, la información de movimiento se refina en forma adicional al nivel de sub-PU con los vectores de movimiento de CU

derivados como los puntos de partida.

[0114] El concepto de emparejamiento bilateral se ilustra en la FIG. 8. El emparejamiento bilateral se utiliza para derivar la información de movimiento mediante la búsqueda del emparejamiento más cercano entre dos bloques a lo largo de la trayectoria de movimiento del bloque actual en dos imágenes de referencia diferentes. Bajo el supuesto de la trayectoria de movimiento continuo, los vectores de movimiento MV1 y MV0 que apuntan a los dos bloques de referencia son proporcionales a las distancias temporales, es decir, TD0 y TD1, entre la imagen actual y las dos imágenes de referencia. Como un caso especial, cuando la imagen actual es temporal entre las dos imágenes de referencia y la distancia temporal entre la imagen actual de las dos imágenes de referencia es igual, el emparejamiento bilateral se convierte en MV bidireccional con base de espejo.

[0115] El emparejamiento de plantilla como se ilustra en la FIG. 9 se utiliza para derivar la información de movimiento al encontrar el emparejamiento más cercano entre una plantilla (bloques contiguos superiores y/o izquierdos del bloque actual) en la imagen actual y un bloque (del mismo tamaño que la plantilla) en una imagen de referencia.

[0116] Predicción del vector de movimiento plano

[0117] La predicción del vector de movimiento plano se consigue mediante el promedio de una interpolación lineal horizontal y vertical en una base de bloque de 4x4 como sigue.

$$P(x,y) = (H \times P_h(x,y) + W \times P_v(x,y) + H \times W)/(2 \times H \times W)$$

[0118] *W* y *H* denotan el ancho y la altura del bloque. (*x, y*) son las coordenadas de subbloque actual con respecto al subbloque de la esquina izquierda anterior. Todas las distancias se indican mediante las distancias entre píxeles dividido por 4. $P(x,y)$ es el vector de movimiento del subbloque actual.

[0119] La predicción horizontal $P_h(x,y)$ y la predicción vertical, $P_v(x,y)$ para la localización (x, y) se calcula como sigue:

$$P_h(x,y) = (W - 1 - x) \times L(-1,y) + (x+1) \times R(W,y)$$

$$P_v(x,y) = (H-1-y)\times A(x,-1) + (y+1)\times B(x,H)$$

[0120] donde $L(-1,y)$ y $R(W,y)$ son los vectores de movimiento de los bloques de 4x4 a la izquierda y derecha del bloque actual. $A(x,-1)$ y $B(x,H)$ son los vectores de movimiento de los bloques de 4x4 arriba y abajo del bloque actual, como se muestra en la FIG. 9.

[0121] La información del movimiento de referencia de los bloques contiguos de la columna izquierda y fila superior se derivan de bloques contiguos espaciales del bloque actual.

[0122] La información del movimiento de referencia de los bloques contiguos de la columna derecha y fila inferior se derivan como sigue.

1) Derivar la información de movimiento del bloque de *4x4* contiguo temporal derecho inferior
2) Calcular los vectores de movimiento de los bloques de *4x4* contiguos de la columna derecha, utilizando la información de movimiento derivada del bloque de *4x4* contiguo inferior derecho junto con la información de movimiento del bloque de *4x4* contiguo superior derecho, como $R(W,y) = ((H-y-1)\times AR + (y+1)\times BR)/H$.
3) Calcular los vectores de movimiento de los bloques de *4x4* contiguos de la fila inferior, utilizando la información de movimiento derivada del bloque de *4x4* contiguo inferior derecho junto con la información de movimiento del bloque de *4x4* contiguo inferior izquierdo, como $B(x,H) = ((W-x-1)\times BL + (x+1)\times BR)/W$.

[0123] donde *AR* es el vector de movimiento del bloque de *4x4* contiguo espacial superior derecho, BR es el vector de movimiento del bloque de *4x4* contiguo temporal inferior derecho, y BL es el vector de movimiento del bloque de *4x4* contiguo espacial inferior izquierdo, como se muestra en la FIG. 10.

[0124] La información de movimiento obtenida de los bloques contiguos para cada lista se escala a la primera imagen de referencia para una lista dada.

[0125] Inserción de lista de candidatos

[0126] Si están disponibles, los candidatos de predicción del movimiento del nivel de sub-PU se insertan en la lista en el siguiente orden predeterminado: Afín→

ATMVP→STMVP→PMMVD→Plano. El orden predeterminado puede estar predefinido o señalado con base en el tipo segmento, capa temporal, modelo de movimiento afín, y/o disponibilidad de predicción de movimiento temporal. El orden predeterminado también puede diferir según el tipo del bloque, forma del bloque, o tamaño del bloque. El número máximo de los candidatos predicción del movimiento del nivel de sub-PU está determinado por un número predefinido y/o el número total de candidatos de predicción del movimiento del nivel de sub-PU disponible como se indica por la sintaxis de alto nivel. En un ejemplo, si el número predefinido es 3, y todos los candidatos de afín, ATMVP, STMVP, PMMVD y Plano están disponibles, entonces el número máximo es de 3. Pero si sólo están disponibles el afín y ATMVP, entonces el número máximo es de 2. Si el número máximo es de 0, entonces se infiere que is_sub_pu_motion_flag es falso ('0') y no se señaliza en el flujo de bits.

[0127] También se puede utilizar otro orden de candidatos por defecto.

En un ejemplo, Afín→PMMVD→ATMVP→STMVP→Plano.

En otro ejemplo, PMMVD→Afín→ATMVP→STMVP→Plano.

En otro ejemplo, pueden utilizarse dos o más candidatos afines. Afín1→Afín2→PMMVD→ATMVP→STMVP→Plano

Afín1→ATMVP→Afín2→Plano

[0128] Reordenamiento de candidato priorizado

[0129] La lista de candidatos de predicción del movimiento del nivel de sub-PU por defecto puede reordenarse en función de sus apariciones en los bloques codificados contiguos. En un ejemplo, el candidato que tiene más ocurrencias de los bloques codificados contiguos se pone en una posición de índice más bajo en la lista.

[0130] Reordenamiento de candidato con prioridad parcial

[0131] Con el fin de reducir la complejidad del reordenamiento de candidatos, el reordenamiento de candidatos con prioridad se aplica sólo a una o varias sublistas. Por ejemplo, el candidato 1-2, 3-4 en el orden predeterminado se reordena por separado en función de su aparición en los bloques codificados contiguos.

[0132] Depuración

[0133] Para reducir la complejidad, no se puede aplicar depuración o se puede aplicar

una depuración parcial en la lista de candidatos de sub-PU. En un ejemplo, la depuración entre ATMVP, STMVP, y plano se puede aplicar, pero no se aplica ninguna depuración entre afín y el resto de los candidatos de sub-PU.

[0134] En otro ejemplo, la depuración solamente se aplica al número de sub-PU si es menor o igual que un valor predefinido.

[0135] Métodos de señalización alternativa de candidato de predicción del movimiento sub-PU

[0136] La selección del candidato de predicción del movimiento de sub-PU se puede señalizar por las banderas de permiso de acuerdo con el orden de inserción candidato en lugar de un índice. En un ejemplo, si el orden de inserción es Afín→PMMVD→ATMVP→STMVP→Plano, entonces la señalización es como se muestra en la Tabla 2.

[0137] En otro método alternativo, los candidatos de predicción del movimiento de sub-PU se pueden agrupar en varios subgrupos. Por ejemplo, Afín, Plano se agrupan en el subgrupo 1, ATMVP y STMVP se agrupan en el subgrupo 2, y PMMVD es otro subgrupo 3. Una señalización ejemplar se muestra en la Tabla 3. Una sintaxis pmmvd_flag se señaliza primero para indicar si es un candidato PMMVD o no. A continuación, si no es un candidato PMMVD, otro elemento de sintaxis sub_group1_flag se señaliza para indicar si se trata de candidatos de subgrupo 1. Si sub_group1_flag es verdadero, lo que indica que son candidatos de subgrupo 1, affine_flag se señaliza para indicar si es un candidato afín. Si no es un candidato afín, entonces planar_mv_flag se establece como verdadero para indicar que es un candidato plano. Si sub_group1_flag es falso, entonces se señaliza atmvp_flag para indicar si es un candidato ATMVP. Si no es un candidato ATMVP, stmvp_flag se establece como verdadero para indicar que es un candidato STMVP.

Tabla 2

	Descriptor
affine_flag	ae(v)
if(!affine_flag) {	
pmmvd_flag	ae(v)
if(!pmmd_flag) {	

atmvp_flag	ae(v)
if(!atmvp_flag) {	
stmvp_flag	ae(v)
if(!stmvp_flag) {	
planar_mv_flag = true	
}	
}	
}	
}	

Tabla 3

	Descriptor
pmmvd_flag	ae(v)
if(!pmmvd_flag) {	
sub_group1_flag	ae(v)
if(sub_group1_flag) {	
affine_flag	ae(v)
if(!affine_flag) {	
planar_mv_flag = true	
}	
} else {	
atmvp_flag	ae(v)
if(!atmvp_flag) {	
stmvp_flag = true	
}	
}	
}	

[0138] Cabe destacar que en la Tabla 2 y Tabla 3, se omite la verificación de disponibilidad por propósitos de facilidad de ejemplo. Una bandera se infiere que es falsa ('0') si el candidato correspondiente no está disponible. Asimismo, cabe destacar que también se pueden aplicar otras variaciones de agrupamiento.

[0139] La FIG. 11 es un diagrama de bloques que ilustra un codificador de video 200 de ejemplo que puede realizar las técnicas de la presente divulgación. La FIG. 11 se proporciona para fines de explicación y no debe considerarse limitante de las técnicas tan ampliamente ejemplificadas y descritas en esta divulgación. Para fines de explicación, esta divulgación describe el codificador de video 200 en el contexto de los

estándares de codificación de video, tal como el estándar de codificación de video HEVC H.266 y el estándar de codificación de video en desarrollo. Sin embargo, las técnicas de la presente divulgación no se limitan a estos estándares de codificación de video, y son aplicables en general a la codificación y decodificación de video.

[0140] En el ejemplo de la FIG. 11, el codificador de video 200 incluye una memoria de datos de video 230, unidad de selección de modo 202, unidad de generación residual 204, unidad de procesamiento de transformada 206, unidad de cuantificación 208, unidad de cuantificación inversa210, unidad de procesamiento de transformada inversa 212, unidad de reconstrucción 214, unidad de filtro 216, memoria intermedia de imágenes decodificadas (DPB) 218, y unidad de codificación por entropía 220.

[0141] La memoria de datos de video 230 puede almacenar datos de video a codificar 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). DPB 218 puede actuar como una memoria de imágenes de referencia que almacena datos de video de referencia para su uso en la predicción de los datos de video posteriores por el codificador de video 200. La memoria de datos video 230 y DPB 218 pueden estar formados por cualquiera de una variedad de dispositivos de memoria, tal como memoria dinámica de acceso aleatorio (DRAM), incluyendo DRAM síncrona (SDRAM), memoria RAM magnetorresistiva (MRAM), RAM resistiva (RRAM), u otros tipos de dispositivos de memoria. La memoria de datos de video 230 y la DPB 218 pueden proporcionarse por el mismo dispositivo de memoria o dispositivos de memoria separados. En diversos ejemplos, la memoria de datos de video 230 puede estar en chip con otros componentes del decodificador de video 200, como se ilustra, o fuera del chip con relación a esos componentes.

[0142] En esta divulgación, la referencia a la memoria de datos de video 230 no debe interpretarse como limitada a una memoria interna para el codificador de video 200, a menos que se describa específicamente como tal, o a una 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 se debe entender como una referencia a una memoria que almacena datos de video que el codificador de video

200 recibe para la codificación (por ejemplo, datos de video para un bloque actual a codificar). La memoria 106 de la FIG. 1 también puede proporcionar almacenamiento temporal de las salidas de las distintas unidades del codificador de video 200.

[0143] Las diversas unidades de la FIG. 11 se ilustran para asistir a la comprensión de 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 funcionalidad en particular, que está predefinida en las operaciones que se pueden realizar. Los circuitos programables se refieren a circuitos que pueden programarse para realizar varias tareas, y proporcionar una funcionalidad flexible en las operaciones que se pueden realizar. Por ejemplo, los circuitos programables pueden ejecutar software o firmware que hacen a los circuitos programables operar en la forma definida por instrucciones del software o firmware. Los circuitos de función fija pueden ejecutar instrucciones de software (por ejemplo, para recibir parámetros o dar salida a parámetros), pero los tipos de operaciones que los circuitos de función fija realizan generalmente son inmutables. En algunos ejemplos, la unidad o las unidades pueden ser bloques de circuitos distintos (de función fija o programable), y en algunos ejemplos, la unidad o las unidades pueden ser circuitos integrados.

[0144] El codificador de video 200 puede incluir unidades aritméticas lógicas (ALU), unidades de funciones elementales (FESU), circuitos digitales, circuitos analógicos, y/o núcleos programables, formados a partir de circuitos programables. En ejemplos donde las operaciones del codificador de video 200 se realizan usando 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 (que no se muestra) puede almacenar dichas instrucciones.

[0145] 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 unidad de selección de modo 202. Los datos de

video en la memoria de datos de video 230 pueden ser datos de video en bruto a codificar.

[0146] La unidad de selección de modo 202 incluye una unidad de estimación de movimiento 222, unidad de compensación de movimiento 224, y una unidad intrapredicción 226. La unidad de selección de modo 202 puede incluir unidades funcionales adicionales para realizar la predicción de video de acuerdo con otros modos de predicción. Como ejemplos, la unidad de selección de modo 202 puede incluir una unidad de paleta, una unidad de copia intrabloque (que puede ser parte de la unidad de estimación de movimiento 222 y/o unidad de compensación de movimiento 224), una unidad afín, una unidad de modelo lineal (LM), o similar.

[0147] La unidad de selección de modo 202 generalmente coordina pasos de codificación múltiple para probar combinaciones de parámetros de codificación y dando como resultado valores de distorsión de tasa para tales combinaciones. Los parámetros de codificación pueden incluir la partición de CTU en CU, los modos de predicción para la CU, tipos de transformada para los datos residuales de la CU, parámetros de cuantificación para los datos residuales de la CU, y etc. La unidad de selección de modo 202 puede en última instancia seleccionar la combinación de parámetros de codificación que tienen valores de distorsión de tasa que son mejores que las otras combinaciones probadas.

[0148] El codificador de video 200 puede particionar 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 210 puede particionar una CTU de la imagen de acuerdo con una estructura de árbol, tal como la estructura QTBT o la estructura en árbol cuádruple de HEVC descrita anteriormente. Como se describió anteriormente, el codificador de video 200 puede formar una o más CU desde la partición de una CTU de acuerdo con la estructura de árbol. Una CU de este tipo también puede denominarse generalmente “bloque de video” o “bloque”.

[0149] En general, la unidad de selección de modo 202 también controla los componentes del mismo (por ejemplo, la unidad de estimación de movimiento 222, unidad de compensación de movimiento 224, y la unidad de intrapredicción 226) para

generar un bloque de predicción para un bloque actual (por ejemplo, una CU actual, o en HEVC, la porción de solapamiento 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 se emparejan estrechamente en una o más imágenes de referencia (por ejemplo, una o más imágenes previamente codificadas almacenadas en DPB 218). En particular, la unidad de estimación de movimiento 222 puede calcular un valor representativo de la similitud de un bloque de referencia potencial con el bloque actual, por ejemplo, de acuerdo con la suma de la diferencia absoluta (SAD), suma de las diferencias al cuadrado (SSD), diferencia media absoluta (MAD), diferencias de medias al cuadrado (MSD), o similares. La unidad de estimación de movimiento 222 puede generalmente realizar estos cálculos usando diferencias muestra a muestra entre el bloque actual y el bloque de referencia considerados. 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 se empareja más estrechamente al bloque actual.

[0150] La unidad de estimación de movimiento 222 puede formar uno o más vectores de movimiento (MV) que definen las posiciones de los bloques de referencia en las imágenes de referencia con relación a la posición del bloque actual en una imagen actual. La unidad de estimación de movimiento 222 puede entonces proporcionar los vectores de movimiento a la unidad de compensación de movimiento 224. Por ejemplo, para la interpredicción unidireccional, la unidad de estimación de movimiento 222 puede proporcionar un solo vector de movimiento, mientras que para la interpredicción bidireccional, la unidad de estimación de movimiento 222 puede proporcionar dos vectores de movimiento. La unidad de compensación de movimiento 224 puede entonces generar un bloque de predicción utilizando los vectores de movimiento. Por ejemplo, la unidad de compensación de movimiento 224 puede recuperar los datos del bloque de referencia usando el vector de movimiento. Como otro ejemplo, si el vector de movimiento tiene precisión de la muestra fraccionada, la unidad de compensación de movimiento 224 puede interpolar valores para el bloque

de predicción de acuerdo con uno o más filtros de interpolación. Además, para la interpredicción bidireccional, la unidad de compensación de movimiento 224 puede recuperar datos de dos bloques de referencia identificados por los vectores de movimiento respectivos y combinar los datos recuperados, por ejemplo, a través de un promedio de muestra por muestra o de un promedio ponderado.

[0151] Como otro ejemplo, para la intrapredicción, o la codificación intrapredicción, la unidad de intrapredicción 226 puede generar el bloque de predicción a partir de muestras contiguas del bloque actual. Por ejemplo, para los modos de dirección, la unidad de intrapredicción 226 puede generalmente de forma matemática combinar los valores de las muestras contiguas y poblar estos valores calculados en la dirección definida a través del bloque actual para producir el bloque de predicción. Como otro ejemplo, para el modo de DC, la unidad de intrapredicción 226 puede calcular un promedio de las muestras contiguas al bloque actual y generar el bloque de predicción para incluir este promedio resultante para cada muestra del bloque de predicción.

[0152] 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 en bruto, no codificada, del bloque actual de la memoria de datos de video 230 y el bloque de predicción de la unidad de selección de modo 202. La unidad de generación residual 204 calcula las diferencias muestra a muestra entre el bloque actual y el bloque de predicción. Las diferencias de muestra 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 usando modulación por codificación de impulsos diferencial residual (RDPCM). En algunos ejemplos, la unidad de generación residual 204 se puede formar usando uno o más circuitos sustractivos que realizan la sustracción binaria.

[0153] En ejemplos donde la unidad de selección de modo 202 particiona CU en PU, cada PU puede estar asociada con una unidad de predicción de luminancia y unidades de predicción de crominancia correspondientes. El codificador de video 200 y el decodificador de video 300 pueden soportar las PU que tienen diversos tamaños.

Como se indicó anteriormente, el tamaño de una CU puede referirse al tamaño del bloque de codificación de luminancia de la CU y el tamaño de una PU puede referirse al tamaño de una unidad de predicción de luminancia de la PU. Suponiendo que el tamaño de una CU particular es 2Nx2N, el codificador de video 200 puede soportar tamaños de PU de 2Nx2N o NxN para intrapredicción y tamaños de PU simétricas de 2Nx2N, 2NxN, Nx2N, NxN, o similar para interpredicción. El codificador de video 200 y decodificador de video 300 también pueden apoyar la partición asimétrica para tamaños de PU de 2NxnU, 2NxnD, nLx2N, y nRx2N para interpredicción.

[0154] En ejemplos donde la unidad de selección de modo 202 no particiona en forma adicional una CU en PU, cada CU puede estar asociada con un bloque de codificación de luminancia y bloques de codificación de crominancia correspondientes. De acuerdo con lo anterior, el tamaño de una CU puede referirse al tamaño del bloque de codificación de luminancia de la CU. El codificador de video 200 y decodificador de video 300 pueden soportar tamaños de CU de 2Nx2N, 2NxN, o Nx2N.

[0155] Para otras técnicas de codificación de video, tal como una codificación de modo de copia intrabloque, una codificación en modo afín, y codificación de modo de modelo lineal (LM), como algunos ejemplos, la unidad de selección de modo 202, a través de las respectivas unidades asociadas con las técnicas de codificación, genera un bloque de predicción para el bloque actual que se codifica. En algunos ejemplos, tal como codificación en modo de paleta, la unidad de selección de modo 202 no puede generar un bloque de predicción, y en cambio generar elementos de sintaxis que indican la manera para reconstruir el bloque basado en una paleta seleccionada. En tales modos, la unidad de selección de modo 202 puede proporcionar estos elementos de sintaxis a la unidad de codificación por entropía 220 a codificar.

[0156] Como se describió anteriormente, 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 un bloque residual para el bloque actual. Para generar el bloque residual, la unidad de generación residual 204 calcula diferencias muestra a muestra entre el bloque de predicción y el bloque actual. Así,

[0157] La unidad de procesamiento de transformada 206 aplica una o más

transformadas al bloque residual para generar un bloque de coeficientes de transformada (denominado en la presente como un “bloque de coeficiente de transformada”). La unidad de procesamiento de transformada 206 puede aplicar varias transformadas a un bloque residual para formar el bloque de coeficientes de transformada. Por ejemplo, la unidad de procesamiento de transformada 206 puede aplicar una transformada de coseno discreta (DCT), una transformada direccional, una transformada de Karhunen-Loeve (KLT), o una transformada conceptualmente similar a un bloque residual. En algunos ejemplos, la unidad de procesamiento de transformada 206 puede realizar varias transformadas a un bloque residual, por ejemplo, una transformada principal y una transformada secundaria, tal como una transformada de rotación. En algunos ejemplos, la unidad de procesamiento de transformada 206 no aplica transformadas a un bloque residual.

[0158] La unidad de cuantificación 208 puede cuantificar los coeficientes de transformada en un bloque de coeficiente de transformada para producir un bloque de coeficientes de transformada cuantificados. La unidad de cuantificación 208 puede cuantificar los coeficientes de transformada de un bloque de coeficientes de transformada de acuerdo con un parámetro de cuantificación (QP) asociado con el bloque actual. El codificador de video 200 (por ejemplo, mediante la unidad de selección de modo 202) puede ajustar el grado de cuantificación aplicado a los bloques de coeficientes 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 transformada cuantificados pueden tener una precisión inferior a los coeficientes de transformada originales producidos por la unidad de procesamiento de transformada 206.

[0159] La unidad de cuantificación inversa 210 y unidad de procesamiento de transformada inversa 212 pueden aplicar cuantificación inversa y transformada inversa a un bloque de coeficientes de transformada cuantificados, respectivamente, para reconstruir un bloque residual del bloque de coeficientes de transformada. La unidad de reconstrucción 214 puede producir un bloque reconstruido correspondiente al bloque actual (aunque potencialmente con algún grado de distorsión) con base en el

bloque reconstruido residual y un bloque de predicción generado por la unidad de selección de modo 202. Por ejemplo, la unidad de reconstrucción 214 puede añadir muestras del bloque residual reconstruido a muestras correspondientes del bloque de predicción generado por la unidad de selección de modo 202 para producir el bloque reconstruido.

[0160] La unidad de filtro 216 puede realizar una o más operaciones de filtro en los bloques reconstruidos. Por ejemplo, la unidad de filtro 216 puede realizar las operaciones de desbloqueo para reducir los artefactos de formación de bloques a lo largo de los bordes de las CU. Las operaciones de la unidad de filtro 216 se pueden omitir, en algunos ejemplos.

[0161] El codificador de video 200 almacena bloques reconstruidos en DPB 218. Por ejemplo, en casos donde no se requieren operaciones de unidad de filtro 216, la unidad de reconstrucción 216 puede almacenar bloques reconstruidos a DPB 218. En ejemplos donde se requieren operaciones de unidad de filtro 224, la unidad de filtro 216 puede almacenar los bloques reconstruidos filtrados a DPB 218. La unidad de estimación de movimiento 222 y la unidad de compensación de movimiento 224 pueden recuperar una imagen de referencia de DPB 218, formada a partir de los bloques reconstruidos (y potencialmente filtrados), para interpredecir bloques de imágenes posteriormente codificados. Además, la unidad de intrapredicción 226 puede utilizar bloques reconstruidos en DPB 218 de una imagen actual para intrapredecir otros bloques en la imagen actual.

[0162] En general, la unidad de codificación por entropía 220 puede codificar por entropía elementos de sintaxis recibidos de otros componentes funcionales del codificador de video 200. Por ejemplo, la unidad de codificación por entropía 220 puede codificar por entropía bloques de coeficientes de transformada cuantificados de unidad de cuantificación 208. Como otro ejemplo, la unidad de codificación por entropía 220 puede codificar por entropía elementos de sintaxis (por ejemplo, información de movimiento para interpredicción o información de intramodo para intrapredicción) de la unidad de selección de modo 202. La unidad de codificación por entropía 220 puede realizar una o más operaciones de codificación por entropía en los

elementos de sintaxis, que son otros ejemplos de datos de video, para generar datos codificados por entropía. Por ejemplo, la unidad de codificación por entropía 220 puede realizar una operación de codificación de longitud variable de contexto adaptativo (CAVLC), una operación de CABAC, una operación de codificación de longitud variable a variable (V2V), una operación de codificación aritmética binaria de contexto adaptativo basada en sintaxis (SBAC), una operación de codificación por entropía de partición de probabilidad de intervalos (PIPE), una operación de codificación de Golomb exponencial, u otro tipo de operación de codificación por entropía de los datos. En algunos ejemplos, la unidad de codificación por entropía 220 puede operar en modo de derivación donde los elementos de sintaxis no se codifican por entropía.

[0163] El codificador de video 200 puede dar salida a un flujo de bits que incluye los elementos de sintaxis codificados por entropía necesarios para reconstruir bloques de un segmento o imagen. En particular, la unidad de codificación por entropía 220 puede dar salida al flujo de bits.

[0164] Las operaciones descritas con anterioridad se describen con respecto a un bloque. Tal descripción debe comprenderse como operaciones para un bloque de codificación de luminancia y/o bloques de codificación de crominancia. Como se ha descrito anteriormente, en algunos ejemplos, el bloque de codificación de luminancia y los bloques de codificación de crominancia son componentes de luminancia y crominancia de una CU. En algunos ejemplos, el bloque de codificación de luminancia y los bloques de codificación de crominancia son componentes de luminancia y crominancia de una PU.

[0165] En algunos ejemplos, no se requiere la repetición de las operaciones que se realizan con respecto a un bloque de codificación de luminancia para los bloques de codificación de crominancia. Como un ejemplo, no se requiere la repetición de las operaciones para identificar un vector de movimiento (MV) y la imagen de referencia para un bloque de codificación de luminancia para la identificación de un MV y una imagen de referencia para los bloques de crominancia. Más bien, el MV para el bloque de codificación de luminancia puede ser escalado para determinar el MV para los

bloques de crominancia, 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 luminancia y los bloques de codificación de crominancia.

[0166] 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 configurada para derivar una plantilla de conversión de la tasa de cuadros de imagen (FRUC) usando valores de predicción de píxeles y realizar una técnica de derivación de vectores de movimiento del lado del decodificador utilizando la plantilla derivada. En otro ejemplo, el decodificador de video 300 puede estar configurado para determinar los vectores de movimiento respectivos de uno o más bloques contiguos de datos de video y derivar un vector de movimiento para un bloque actual de datos de video usando los vectores de movimiento respectivos de los uno o más bloques contiguos de datos de video. En otro ejemplo, el decodificador de video 300 puede estar configurado para determinar los vectores de movimiento respectivos de uno o más bloques colocados de datos de video, y derivar un vector de movimiento para un bloque actual de datos de video usando los vectores de movimiento respectivos de los uno o más bloques colocados de datos de video.

[0167] La FIG. 12 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. 12 se proporciona con fines explicativos y no es limitativa en las técnicas tan ampliamente ejemplificadas y descritas en esta divulgación. A título explicativo, la presente divulgación describe el decodificador de video 300 que se detalla de acuerdo con las técnicas de JEM, H.266/VVC, y HEVC. Sin embargo, las técnicas de la presente divulgación pueden realizarse por dispositivos de codificación de video que están configurados para otros estándares de codificación de video.

[0168] En el ejemplo de la FIG. 12, el decodificador de video 300 incluye memoria intermedia de imagen codificada (CPB) 320, unidad de decodificación por 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 memoria intermedia de imagen decodificada (DPB) 314. la unidad de procesamiento de predicción 304 incluye una unidad de compensación de movimiento 316 y 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 mayor o menor cantidad o diferentes componentes funcionales.

[0169] La memoria CPB 320 puede almacenar datos de video, tal 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 CPB 320 se pueden obtener, por ejemplo, de un medio legible por computadora 110 (FIG. 1). La memoria CPB 320 puede incluir una CPB que almacena datos de video codificados (por ejemplo, elementos de sintaxis) de un flujo de bits de video codificado. También, la memoria CPB 320 puede almacenar datos de video distintos de elementos de sintaxis de una imagen codificada, como los datos temporales que representan las salidas de las distintas unidades del decodificador de video 300. DPB 314 generalmente almacena imágenes decodificadas, que el decodificador de video 300 puede emitir y/o utilizar como datos de video de referencia cuando se decodifican datos o imágenes posteriores del flujo de bits de video codificado. La memoria CPB 320 y DPB 314 pueden formarse mediante cualquiera de una variedad de dispositivos de memoria, tal como una memoria dinámica de acceso aleatorio (DRAM), que incluye DRAM síncrona (SDRAM), memoria RAM magnetorresistiva (MRAM), RAM resistiva (RRAM), u otros tipos de dispositivos de memoria), u otros tipos de dispositivos de memoria. La memoria CPB 320 y DPB 314 pueden proporcionarse por el mismo dispositivo de memoria o dispositivos de memoria separados. En diversos ejemplos, la memoria CPB 320 puede estar en chip con otros componentes del decodificador de video 300, o fuera del chip con relación a esos componentes.

[0170] Adicional o alternativamente, en algunos ejemplos, el decodificador de video 300 puede recuperar los datos codificados de video de la memoria 120 (FIG. 1). Es decir, la memoria 120 puede almacenar datos como se discutió anteriormente con la memoria CPB 320. Del mismo modo, la memoria 120 puede almacenar instrucciones a ejecutar por el decodificador de video 300, cuando algunas o todas las funcionalidades del decodificador de video 300 se implementan en software para ejecución por circuitos procesamiento del decodificador de video 300.

[0171] Las diferentes unidades mostradas en la FIG. 12 se ilustran para asistir con la comprensión de 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. En forma similar a la FIG. 11, los circuitos de función fija se refieren a circuitos que proporcionan funcionalidad en particular, que está predefinida en las operaciones que se pueden realizar. Los circuitos programables se refieren a circuitos que pueden programarse para realizar varias tareas, y proporcionar una funcionalidad flexible en las operaciones que se pueden realizar. Por ejemplo, los circuitos programables pueden ejecutar software o firmware que hacen a los circuitos programables operar en la forma definida por instrucciones del software o firmware. Los circuitos de función fija pueden ejecutar instrucciones de software (por ejemplo, para recibir parámetros o dar salida a parámetros), pero los tipos de operaciones que los circuitos de función fija realizan generalmente son inmutables. En algunos ejemplos, las una o más unidades pueden ser bloques de circuitos distintos (de función fija o programable), y en algunos ejemplos, las una o más unidades pueden ser circuitos integrados.

[0172] El decodificador de video 300 puede incluir ALU, EFU, circuitos digitales, circuitos analógicos, y/o núcleos programables formados a partir de circuitos programables. En ejemplos donde las operaciones de decodificador de video 300 se llevan a cabo por el software que se ejecuta en los circuitos programables, la memoria en el chip o fuera del chip puede almacenar instrucciones (por ejemplo, código de objeto) del software que el decodificador de video 300 recibe y ejecuta.

[0173] La unidad de decodificación por entropía 302 puede recibir datos de video

codificados de la CPB y decodificar por entropía datos de video para reproducir elementos de sintaxis. La unidad de procesamiento de predicción 304, unidad de cuantificación inversa 306, unidad de procesamiento de transformada inversa 308, unidad de reconstrucción 310, y unidad de filtro 312 pueden generar datos de video decodificados con base en los elementos de sintaxis extraídos del flujo de bits.

[0174] En general, el decodificador de video 300 reconstruye una imagen sobre una base de bloque por bloque. El decodificador de video 300 puede realizar una operación de reconstrucción en cada bloque individual (donde el bloque actualmente siendo reconstruido, es decir, decodificado, puede denominarse "bloque actual").

[0175] La unidad de decodificación por entropía 302 puede decodificar por entropía elementos de sintaxis que definen de coeficientes de transformada cuantificados de un bloque de coeficientes de transformada cuantificados, así como información de transformada, tal como un parámetro de cuantificación (QP) y/o indicación de modo de transformada. La unidad de cuantificación inversa 306 puede utilizar el QP asociado con el bloque de coeficiente de transformada cuantificado para determinar un grado de cuantificación y, así mismo, un grado de cuantificación inversa para la unidad de cuantificación inversa 306 a aplicar. La unidad de cuantificación inversa 306 puede, por ejemplo, realizar una operación de desplazamiento a la izquierda a nivel de bits para la cuantificación inversa de los coeficientes de transformada cuantificados. La unidad de cuantificación inversa 306 puede formar de este modo un bloque de coeficientes de transformada incluyendo coeficientes de transformada.

[0176] Después de que la unidad de cuantificación inversa 306 forma el bloque de coeficiente de transformada, la unidad de procesamiento de transformada inversa 308 puede aplicar una o más transformada inversa al bloque de coeficiente de transformada para generar un bloque residual asociado con el bloque actual. Por ejemplo, la unidad de procesamiento de transformada inversa 308 puede aplicar una DCT inversa, una transformada inversa de número entero, una transformada inversa Karhunen-Loeve (KLT), una transformada inversa de rotación, una transformada inversa direccional, u otra transformada inversa al bloque de coeficientes.

[0177] Además, la unidad de procesamiento de predicción 304 genera un bloque de

predicción de acuerdo con elementos de sintaxis de información de predicción decodificados por entropía por la unidad de decodificación por entropía 302. Por ejemplo, si los elementos de sintaxis de la información de predicción indican que el bloque actual está interpredicho, la unidad de compensación de movimiento 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 DPB 314 desde la que recibir un bloque de referencia, así como un vector de movimiento que identificar una ubicación del bloque de referencia en la imagen de referencia con respecto a la ubicación del bloque actual en la imagen actual. La unidad de compensación de movimiento 316 puede generalmente llevar a cabo 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. 11).

[0178] Como otro ejemplo, si los elementos de sintaxis de información de predicción indican que el bloque actual es intrapredicho, la unidad de intrapredicción 318 puede generar el bloque de predicción de acuerdo con un modo de intrapredicción indicado por los elementos de sintaxis de información de predicción. Nuevamente, la unidad de intrapredicción 318 puede generalmente llevar a cabo el proceso de intrapredicción de una manera que es sustancialmente similar a la descrita con respecto a la unidad de intrapredicción 226 (Fig. 11). La unidad de intrapredicción 318 puede recuperar los datos de las muestras contiguas para el bloque actual de DPB 314.

[0179] 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 añadir las muestras del bloque residual a las muestras del bloque de predicción correspondiente para reconstruir el bloque actual.

[0180] La unidad de filtro 312 puede realizar una o más operaciones de filtro en los bloques reconstruidos. Por ejemplo, la unidad de filtro 312 puede realizar las operaciones de desbloqueo para reducir los artefactos de formación de bloques a lo largo de bordes de los bloques reconstruidos. Las operaciones de la unidad de filtro 312 no se realizan necesariamente en todos los ejemplos.

[0181] El decodificador de video 300 puede almacenar los bloques reconstruidos en

DPB 314. Como se discutió anteriormente, DPB 314 puede proporcionar información de referencia, tal como muestras de una imagen actual para intrapredicción e imágenes decodificadas previamente para la compensación de movimiento subsiguiente, a la unidad de proceso de predicción 304. Además, el decodificador de video 300 puede dar salida a imágenes decodificadas de DPB para su presentación posterior en un dispositivo de visualización, tal como el dispositivo de presentación 118 de la FIG. 1.

[0182] De esta manera, el decodificador de video 300 representa un ejemplo de un dispositivo de decodificación 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 implementar los métodos y procesos discutidos en la presente.

[0183] Por ejemplo, los datos de video codificados pueden incluir la siguiente sintaxis:

Tabla 4. Sintaxis de Datos de Fusión

merge_data(x0, y0, cbWidth, cbHeight) {	Descriptor
mmvd_flag[x0][y0]	ae(v)
if(mmvd_flag[x0][y0] = = 1) {	
mmvd_merge_flag[x0][y0]	ae(v)
mmvd_distance_idx[x0][y0]	ae(v)
mmvd_direction_idx[x0][y0]	ae(v)
} else {	
if(MaxNumSubblockMergeCand > 0 && cbWidth >= 8 && cbHeight >= 8)	
merge_subblock_flag[x0][y0]	ae(v)
if(merge_subblock_flag[x0][y0] = = 1) {	
if(MaxNumSubblockMergeCand > 1)	
merge_subblock_idx[x0][y0]	ae(v)
} else {	
if(sps_ciip_enabled_flag && cu_skip_flag[x0][y0] = = 0 && (cbWidth * cbHeight) >= 64 && cbWidth < 128 && cbHeight < 128) {	
ciip_flag[x0][y0]	ae(v)
if(ciip_flag[x0][y0]) {	
if (cbWidth <= 2 * cbHeight | | cbHeight <= 2 * cbWidth)	
ciip_luma_mpm_flag[x0][y0]	ae(v)
if(ciip_luma_mpm_flag[x0][y0])	
ciip_luma_mpm_idx[x0][y0]	ae(v)

}	
}	
if(sps_triangle_enabled_flag && tile_group_type = = B && cbWidth * cbHeight >= 64)	
merge_triangle_flag[x0][y0]	ae(v)
if(merge_triangle_flag[x0][y0])	
merge_triangle_idx[x0][y0]	ae(v)
else if(MaxNumMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
}	
}	
}	

[0184] merge_subblock_flag[x0][y0] especifica si los parámetros de interpredicción basados en subbloques para la unidad de codificación actual se infieren a partir de bloques contiguos. Los índices de matriz x0, y0 especifican la ubicación (x0, y0) de la muestra luma superior izquierda del bloque considerado de codificación respecto a la muestra de luminancia superior izquierda de la imagen. Cuando merge_subblock_flag[x0][y0] no está presente, se infiere que es igual a 0.

[0185] merge_subblock_idx[x0][y0] especifica el índice candidato de fusión de la lista de candidatos de fusión con base en subbloque, donde x0, y0 especifican la ubicación (x0, y0) de la muestra de luminancia superior izquierda del bloque de codificación considerado relativo a la muestra de luminancia superior izquierda de la imagen.

[0186] Cuando merge_subblock_idx[x0][y0] no está presente, se infiere que es igual a 0.

[0187] Si merge_subblock_flag[xCb][yCb]es igual a 1, el proceso de derivación para vectores de movimiento y los índices de referencia en el modo de combinación de subbloque como se especifica en 8.4.4.2 se invocan con la ubicación del bloque de codificación de luminancia (xCb, yCb), el ancho del bloque de codificación de luminancia cbWidth y la altura del bloque de codificación de luminancia cbHeight como entradas, el número de subbloques de codificación de luminancia en dirección horizontal numSbX y en dirección vertical numSbY, los índices de referencia refIdxL0,

refIdxL1, las matrices de bandera de utilización de lista predicción predFlagL0[xSbIdx][ySbIdx] and predFlagL1[xSbIdx][ySbIdx], las matrices de vector de movimiento de subbloque de luminancia mvL0[xSbIdx][ySbIdx] y mvL0[xSbIdx][ySbIdx], y las matrices de vector de movimiento de subbloque de crominancia mvCL0 [xSbIdx] [ySbIdx] y mvCL1 [xSbIdx] [ySbIdx], con xSbIdx = 0.. numSbX – 1, ySbIdx = 0 .. numSbY – 1, y el índice de peso de bipredicción gbiIdx como salidas.

[0188] Cabe reconocer que, dependiendo del ejemplo, ciertos actos o eventos de cualquiera de las técnicas descritas en la presente se pueden realizar en una secuencia diferente, se pueden añadir, combinar, u omitir por completo (por ejemplo, no todos los actos o eventos descritos son necesarios para la práctica de las técnicas). Además, en algunos ejemplos, los actos o eventos pueden realizarse simultáneamente, por ejemplo, a través de procesamiento de múltiples hilos, procesamiento de interrupción, o procesadores múltiples, en lugar de secuencialmente.

[0189] En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, firmware, o cualquier combinación de los mismos. Si se implementan en software, las funciones pueden almacenarse en o transmitirse a través de una o más instrucciones o código en un medio legible por computadora y ejecutarse por una unidad de procesamiento basado en hardware. Los medios legibles por computadora pueden incluir medios de almacenamiento legibles por computadora, lo que corresponde a un medio tangible tal como un medio de almacenamiento de datos, o medios de comunicación incluyendo cualquier medio que facilite la transferencia de un programa informático de un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por computadora generalmente pueden corresponder a (1) medios de almacenamiento legibles por computadora tangibles no transitorios o (2) un medio de comunicación tal como una onda de señal o portador. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que se puede acceder por una o más computadoras o uno o más procesadores para recuperar instrucciones, códigos y/o estructuras de datos para la aplicación de las técnicas descritas en esta revelación. Un producto de

programa informático puede incluir un medio legible por computadora.

[0190] A modo de ejemplo, y no de limitación, tales medios de almacenamiento legibles por computadora pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento de disco óptico, almacenamiento en disco magnético u otros dispositivos de almacenamiento magnético, memoria flash, o cualquier otro medio que puede ser utilizado para almacenar el código del programa deseado en forma de instrucciones o estructuras de datos y que se puede acceder por una computadora. Además, cualquiera se denomina correctamente medio legible por computadora. Por ejemplo, si las instrucciones se transmiten desde un sitio web, servidor, u otra fuente remota usando un cable coaxial, cable de fibra óptica, par trenzado, línea de abonado digital (DSL) o tecnologías inalámbricas tales como infrarrojas, de radio y microondas, entonces el cable coaxial, cable de fibra óptica, par trenzado, DSL o las tecnologías inalámbricas tales como infrarrojas, de radio y microondas se incluyen en la definición de medio. Debe entenderse, sin embargo, que los medios de almacenamiento legibles por computadora y medios de almacenamiento de datos no incluyen las conexiones, ondas portadoras, señales u otros medios transitorios, sino que se dirigen a medios de almacenamiento tangibles no transitorios. *Disk* y *disc*, de acuerdo con lo utilizado en la presente, incluyen disco compacto (CD), disco láser, discos ópticos, discos versátiles digitales (DVD), disquetes y discos Blu-ray, donde *disks* generalmente reproducen datos magnéticamente, mientras que *discs* reproducen datos ópticamente con láseres. Las combinaciones de los anteriores también deben incluirse dentro del alcance de medios legibles por computadora.

[0191] Las instrucciones pueden ejecutarse por uno o más procesadores, tal como uno o más procesadores digitales de señal (DSP), microprocesadores de propósito general, circuitos integrados de aplicación específico (ASIC), matrices lógicas programables en campo (FPGA), u otros circuitos equivalentes integrados o lógicas discretas. Por consiguiente, el término "procesador", de acuerdo con lo utilizado en la presente, puede referirse a cualquiera de las estructuras anteriores o cualquier otra estructura adecuada para la aplicación de las técnicas descritas en la presente. Además, en algunos aspectos, la funcionalidad descrita en la presente puede proporcionarse

dentro de los módulos de hardware y/o software configurados para codificar y decodificar, o incorporar en un códec combinado. Además, las técnicas pueden aplicarse plenamente en uno o más circuitos o elementos lógicos.

[0192] 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 circuitos integrados (por ejemplo, un conjunto de chips). Diversos componentes, módulos o unidades se describen en esta revelación para enfatizar los aspectos funcionales de los dispositivos configurados para llevar a cabo las técnicas reveladas, pero no necesariamente requieren la realización de diferentes unidades de hardware. Más bien, como se ha descrito anteriormente, las diversas unidades pueden combinarse en una unidad de hardware códec o proporcionarse por una colección de unidades de hardware interoperativas, incluyendo uno o más procesadores como se describe anteriormente, en conjunción con el software y/o firmware adecuado.

[0193] Se han descrito varios ejemplos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.