

SELECCIÓN DE TRANSFORMACIÓN IMPLÍCITA EN CODIFICACIÓN DE VIDEO

[0001] La presente solicitud reivindica la prioridad de la solicitud de patente US N.º 16/815.920, presentada el 11 de marzo de 2020, que reivindica el beneficio de la solicitud de patente provisional US N.º 62/817.397, presentada el 12 de marzo de 2019, cuyo contenido completo es incorporado en el presente documento como referencia.

CAMPO TÉCNICO

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

ANTECEDENTES

[0003] Las capacidades de video digital se pueden incorporar en una amplia gama de dispositivos, incluyendo televisores digitales, sistemas de transmisión digital directa, sistemas de transmisión inalámbrica, asistentes digitales personales (PDA), computadoras portátiles o de escritorio, tablets, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos celulares o de radio satelital, los llamados “teléfonos inteligentes”, dispositivos de videoconferencia, dispositivos de transmisión de video de flujo continuo, y similares. Los dispositivos de video digital implementan técnicas de codificación de video, como las descritas en los estándares definidos por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificación de video avanzada (AVC), ITU-T H.265/Codificación de video de alta eficiencia (HEVC) y extensiones de dichos estándares. Los dispositivos de video pueden transmitir, recibir, codificar, decodificar y/o almacenar información de video digital de manera más eficiente mediante la implementación de tales técnicas de codificación de video.

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

bloques vecinos en la misma imagen o predicción temporal con respecto a muestras de referencia en otras imágenes de referencia. Las imágenes pueden denominarse cuadros, y las imágenes de referencia pueden denominarse cuadros de referencia.

SÍNTESIS

[0005] En un ejemplo, un método incluye inferir, para un bloque de transformación actual de un bloque de video actual, un tipo de transformación de una pluralidad de tipos de transformación que incluye una o más transformadas de coseno discretas (DCT) y una o más transformadas de seno discretas (DST), en donde inferir el tipo de transformación comprende: determinar un tamaño del bloque de transformación actual; determinar si el bloque de video actual está particionado mediante particiones intrasub-bloques (ISP); y en respuesta a la determinación de que el tamaño del bloque de transformación actual satisface un umbral de tamaño y que el bloque de video actual está particionado mediante ISP, seleccionar una DST particular de uno o más DST como el tipo de transformación seleccionado; transformar, con el uso del tipo de transformación seleccionado, el bloque de transformación actual para obtener un bloque de datos residuales reconstruidos para el bloque de video; y reconstruir, en función de los datos residuales reconstruidos para el bloque de video, el bloque de video.

[0006] En otro ejemplo, un dispositivo incluye una memoria configurada para almacenar bloques de video; y uno o más procesadores implementados en circuitos y configurados para: inferir, para un bloque de transformación actual de un bloque de video actual, un tipo de transformación de una pluralidad de tipos de transformación que incluye una o más DCT y una o más DST, en donde, para inferir el tipo de transformación, el procesador o los procesadores están configurados para: determinar un tamaño del bloque de transformación actual; determinar si el bloque de video actual está particionado mediante ISP; y seleccionar, en respuesta a la determinación de que el tamaño del bloque de transformación actual satisface un umbral de tamaño y que el bloque de video actual está particionado mediante ISP, una DST particular de uno o más DST como el tipo de transformación seleccionado; transformar, con el uso del tipo de transformación seleccionado, el bloque de transformación actual para obtener un bloque de datos residuales reconstruidos para el bloque de video; y reconstruir, en función de los datos residuales reconstruidos para el bloque de video, el bloque de video.

[0007] En otro ejemplo, un medio de almacenamiento legible por computadora almacena instrucciones que, cuando se ejecutan, hacen que uno o más procesadores de un dispositivo de codificación de video: infieran, para un bloque de transformación actual de un bloque de video actual, un tipo de transformación de una pluralidad de tipos

de transformación que incluye una o más DCT y una o más DST, en donde las instrucciones que hacen que uno o más procesadores infieran el tipo de transformación comprenden instrucciones que hacen que uno o más procesadores: determinen un tamaño del bloque de transformación actual; determinen si el bloque de video actual está particionado mediante ISP; y seleccionen, en respuesta a la determinación de que el tamaño del bloque de transformación actual satisface un umbral de tamaño y que el bloque de video actual está particionado mediante ISP, una DST particular de uno o más DST como el tipo de transformación seleccionado; transformen, con el uso del tipo de transformación seleccionado, el bloque de transformación actual para obtener un bloque de datos residuales reconstruidos para el bloque de video; y reconstruyan, en función de los datos residuales reconstruidos para el bloque de video, el bloque de video.

[0008] Los detalles de uno o más ejemplos de esta divulgación se establecen en los dibujos adjuntos y la descripción a continuación. Otras características, objetos y ventajas de varios aspectos de las técnicas resultarán evidentes a partir de la descripción y los dibujos, y de las reivindicaciones.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

[0009] 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.

[0010] Las FIG. 2A y 2B son diagramas conceptuales que ilustran un ejemplo de estructura de árbol binario *quadtree* (QTBT) y una unidad de árbol de codificación (CTU) correspondiente.

[0011] La FIG. 2C es un diagrama conceptual que ilustra otro ejemplo de estructura de *quadtree* y la unidad de árbol correspondiente.

[0012] La FIG. 3 es un diagrama de bloques que ilustra un codificador de video de ejemplo que puede realizar las técnicas de esta divulgación.

[0013] La FIG. 4 es un diagrama de bloques que ilustra un decodificador de video de ejemplo que puede realizar las técnicas de esta divulgación.

[0014] La FIG. 5 es un diagrama de bloques que ilustra un sistema para codificación de video híbrido con selección de transformación adaptativa.

[0015] La FIG. 6 es un diagrama conceptual que ilustra la implementación de transformadas separables con líneas horizontales y verticales que se transforman de forma independiente.

[0016] La FIG. 7 es un diagrama conceptual que ilustra un bloque de ejemplo para el que un codificador de video puede derivar implícitamente transformaciones, de acuerdo

con una o más técnicas de esta divulgación.

[0017] La FIG. 8 es un diagrama conceptual que ilustra direcciones de intrapredicción.

[0018] La FIG. 9 es un diagrama de flujo que ilustra un método de ejemplo para codificar un bloque actual.

[0019] La FIG. 10 es un diagrama de flujo que ilustra un método de ejemplo para decodificar un bloque actual.

[0020] La FIG. 11 es un diagrama de flujo que ilustra un método de ejemplo para inferir un tipo de transformación para un bloque de transformación de un bloque de video, de acuerdo con una o más técnicas de esta divulgación.

DESCRIPCIÓN DETALLADA

[0021] En general, esta divulgación describe técnicas para la selección de transformación implícita en la codificación de video. Como se describe con más detalle a continuación, después de la predicción, tal como intrapredicción o interpredicción de un bloque, un codificador de video puede calcular datos residuales para el bloque. Los datos residuales, como un bloque residual, representan las diferencias muestra por muestra entre el bloque y un bloque de predicción para el bloque, formado con el uso del modo de predicción correspondiente. El codificador de video puede aplicar una o más transformaciones al bloque residual, para producir datos transformados en un dominio de transformación en lugar del dominio de muestra. Por ejemplo, el codificador de video puede aplicar una transformada de coseno discreta (DCT). En algunos ejemplos, el codificador de video puede utilizar diferentes tipos de transformaciones. Por ejemplo, el codificador de video puede usar varios tipos de DCT.

[0022] Un decodificador de video puede aplicar una transformación inversa al decodificar los datos de video. Cuando el codificador de video puede utilizar diferentes tipos de transformaciones, puede ser necesario que el decodificador de video determine qué transformación fue utilizada por el codificador de video. En algunos ejemplos, el codificador de video puede señalar explícitamente (por ejemplo, codificar un elemento de sintaxis con un valor que indique) qué tipo de transformación se utilizó al transformar los datos residuales. Sin embargo, en algunos ejemplos, puede que no sea deseable señalar explícitamente el tipo de transformación utilizado (por ejemplo, debido a la sobrecarga de señalización).

[0023] De acuerdo con una o más técnicas de esta divulgación, el decodificador de video puede determinar implícitamente qué tipo de transformación se usó al transformar los datos residuales. Por ejemplo, el decodificador de video puede aplicar un conjunto de reglas para determinar qué tipo de transformación se utilizó al transformar los datos

residuales en base a la información secundaria disponible en el decodificador de video (por ejemplo, ya sea explícitamente señalizada o implícitamente derivada de información señalizada). El codificador de video puede aplicar las mismas reglas al determinar qué tipo de transformación usar. Como tal, el codificador de video y el decodificador de video pueden determinar qué tipo de transformación utilizar sin señalización explícita del tipo de transformación.

[0024] La FIG. 1 es un diagrama de bloques que ilustra un ejemplo de sistema de codificación y decodificación de video 100 que puede realizar las técnicas de la presente divulgación. Las técnicas de la presente divulgación están generalmente dirigidas a la codificación (codificación y/o decodificación) de datos de video. En general, los datos de video incluyen cualquier dato para procesar un video. Por lo tanto, los datos de video pueden incluir video sin procesar y sin codificar, video codificado, video decodificado (por ejemplo, reconstruido) y metadatos de video, como datos de señalización.

[0025] Como se muestra en la FIG. 1, el sistema 100 incluye un dispositivo de origen 102 que proporciona datos de video codificados para ser decodificados y visualizados por un dispositivo de destino 116, en este ejemplo. En particular, el dispositivo de origen 102 proporciona los datos de video al dispositivo de destino 116 a través de un medio legible por computadora 110. El dispositivo de origen 102 y el dispositivo de destino 116 pueden comprender cualquiera de una amplia gama de dispositivos, incluyendo computadoras de escritorio, computadoras portátiles (es decir, computadoras portátiles), tablets, decodificadores, microteléfonos como teléfonos inteligentes, televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de transmisión de video de flujo continuo o similares. En algunos casos, el dispositivo de origen 102 y el dispositivo de destino 116 pueden estar equipados para la comunicación inalámbrica y, por lo tanto, pueden denominarse dispositivos de comunicación inalámbrica.

[0026] En el ejemplo de la FIG. 1, el dispositivo de origen 102 incluye la fuente de video 104, la memoria 106, el codificador de video 200 y la interfaz de salida 108. El dispositivo de destino 116 incluye la interfaz de entrada 122, el decodificador de video 300, la memoria 120 y el dispositivo de visualización 118. De acuerdo con esta divulgación, el codificador de video 200 del dispositivo de origen 102 y el decodificador de video 300 del dispositivo de destino 116 pueden configurarse para aplicar las técnicas para la selección de transformación implícita. Por lo tanto, el dispositivo de origen 102 representa un ejemplo de un dispositivo de codificación de video, mientras que el dispositivo de destino 116 representa un ejemplo de un dispositivo de decodificación de

video. En otros ejemplos, un dispositivo de origen y un dispositivo de destino pueden incluir otros componentes o disposiciones. Por ejemplo, el dispositivo de origen 102 puede recibir datos de video desde una fuente de video externa, tal como una cámara externa. Del mismo modo, el dispositivo de destino 116 puede interactuar con un dispositivo de visualización externo, en lugar de incluir un dispositivo de visualización integrado.

[0027] El sistema 100 como se muestra en la FIG. 1 es simplemente un ejemplo. En general, cualquier dispositivo de codificación y/o decodificación de video digital puede realizar las técnicas para la selección de transformación implícita. El dispositivo de origen 102 y el dispositivo de destino 116 son simplemente ejemplos de tales dispositivos de codificación en los que el dispositivo de origen 102 genera datos de video codificados para su transmisión al dispositivo de destino 116. Esta divulgación se refiere a un dispositivo de “codificación” como un dispositivo que realiza la codificación (codificación y/o decodificación) de datos. Por lo tanto, el codificador de video 200 y el decodificador de video 300 representan ejemplos de dispositivos de codificación, en particular, un codificador de video y un decodificador de video, respectivamente. En algunos ejemplos, los dispositivos 102 y 116 pueden funcionar de una manera sustancialmente simétrica de manera que cada uno de los dispositivos 102 y 116 incluye componentes de codificación y decodificación de video. Por lo tanto, el sistema 100 puede admitir la transmisión de video unidireccional o bidireccional entre los dispositivos 102 y 116, por ejemplo, para transmisión de video de flujo continuo, reproducción de video, transmisión de video o telefonía de video.

[0028] En general, la fuente de video 104 representa una fuente de datos de video (es decir, datos de video sin codificar) y proporciona una serie secuencial de imágenes (también denominadas “cuadros”) de los datos de video al codificador de video 200, que codifica datos para las imágenes. La fuente de video 104 del dispositivo de origen 102 puede incluir un dispositivo de captura de video, tal como una cámara de video, un archivo de video que contiene video sin procesar previamente capturado, y/o una interfaz de alimentación de video para recibir video de un proveedor de contenido de video. Como alternativa adicional, la fuente de video 104 puede generar datos basados en gráficos de computadora como el video fuente, o una combinación de video en vivo, video archivado y video generado por computadora. En cada caso, el codificador de video 200 codifica los datos de video capturados, capturados previamente o generados por computadora. El codificador de video 200 puede reorganizar las imágenes del orden recibido (a veces denominado “orden de visualización”) en un orden de codificación para

la codificación. El codificador de video 200 puede generar un flujo de bits que incluye datos de video codificados. El dispositivo de origen 102 puede entonces emitir los datos de video codificados a través de la interfaz de salida 108 en un medio legible por computadora 110 para recepción y/o recuperación por, por ejemplo, la interfaz de entrada 122 del dispositivo de destino 116.

[0029] La memoria 106 del dispositivo de origen 102 y la memoria 120 del dispositivo de destino 116 representan memorias de uso general. En cierto ejemplo, las memorias 106, 120 pueden almacenar datos de video sin procesar, por ejemplo, video sin procesar de la fuente de video 104 y datos de video sin procesar decodificados del decodificador de video 300. Adicional o alternativamente, las memorias 106, 120 pueden almacenar instrucciones de *software* ejecutables, por ejemplo, por el codificador de video 200 y decodificador de video 300, respectivamente. Aunque se muestran por separado del codificador de video 200 y el decodificador de video 300 en este ejemplo, debe entenderse que el codificador de video 200 y el decodificador de video 300 también pueden incluir memorias internas para propósitos funcionalmente similares o equivalentes. Además, las memorias 106, 120 pueden almacenar datos de video codificados, por ejemplo, la salida del codificador de video 200 y la entrada al decodificador de video 300. En algunos ejemplos, partes de las memorias 106, 120 pueden asignarse como uno o más búferes de video, por ejemplo, para almacenar datos de video sin procesar, decodificados y/o codificados.

[0030] El medio legible por computadora 110 puede representar cualquier tipo de medio o dispositivo capaz de transportar los datos de video codificados desde el dispositivo de origen 102 al dispositivo de destino 116. En un ejemplo, el medio legible por computadora 110 representa un medio de comunicación para habilitar el dispositivo de origen 102 para transmitir datos de video codificados directamente al dispositivo de destino 116 en tiempo real, por ejemplo, a través de una red de radiofrecuencia o una red informática. La interfaz de salida 108 puede modular una señal de transmisión que incluye los datos de video codificados, y la interfaz de entrada 122 puede 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 inalámbrico o cableado, como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión física. El medio de comunicación puede formar parte de una red basada en paquetes, como una red de área local, una red de área amplia o una red global como Internet. El medio de comunicación puede incluir enrutadores, conmutadores, estaciones base o cualquier

otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo de origen 102 al dispositivo de destino 116.

[0031] En algunos ejemplos, el dispositivo de origen 102 puede emitir datos codificados desde la interfaz de salida 108 al dispositivo de almacenamiento 116. De manera similar, el dispositivo de destino 116 puede acceder a los datos codificados desde el dispositivo de almacenamiento 116 a través de la interfaz de entrada 122. El dispositivo de almacenamiento 116 puede incluir cualquiera de una variedad de medios de almacenamiento de datos distribuidos o de acceso local, como un disco duro, Discos Blu-ray, DVD, CD-ROM, memoria flash, memoria volátil o no volátil, o cualquier otro medio de almacenamiento digital adecuado para almacenar datos de video codificados.

[0032] En algunos ejemplos, el dispositivo de origen 102 puede enviar datos de video codificados al servidor de archivos 114 u otro dispositivo de almacenamiento intermedio que puede almacenar video codificado generado por el dispositivo de origen 102. El dispositivo de destino 116 puede acceder a los datos de video almacenados desde el servidor de archivos 114 mediante transmisión de flujo continuo o descarga. El servidor de archivos 114 puede ser cualquier tipo de dispositivo de servidor capaz de almacenar datos de video codificados y transmitir esos datos de video codificados al dispositivo de destino 116. El servidor de archivos 114 puede representar un servidor web (por ejemplo, para un sitio web), un servidor de protocolo de transferencia de archivos (FTP), un dispositivo de red de entrega de contenido o un dispositivo de almacenamiento conectado a la red (NAS). El dispositivo de destino 116 puede acceder a datos de video codificados desde el servidor de archivos 114 a través de cualquier conexión de datos estándar, incluida una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión Wi-Fi), una conexión por cable (por ejemplo, DSL, módem de cable, etc.) o una combinación de ambos que sea adecuada para acceder a los datos de video codificados almacenados en el servidor de archivos 114. El servidor de archivos 114 y la interfaz de entrada 122 pueden configurarse para funcionar de acuerdo con un protocolo de transmisión de flujo continuo, un protocolo de transmisión de descarga o una combinación de estos.

[0033] La interfaz de salida 108 y la interfaz de entrada 122 pueden representar transmisores/receptores inalámbricos, módems, componentes de red cableados (por ejemplo, tarjetas Ethernet), componentes de comunicación inalámbrica que funcionan de acuerdo con cualquiera de una variedad de estándares IEEE 802.11 u otros componentes físicos. En los ejemplos en los que la interfaz de salida 108 y la interfaz de entrada 122 comprenden componentes inalámbricos, la interfaz de salida 108 y la

interfaz de entrada 122 se pueden configurar para transferir datos, como datos de video codificados, de acuerdo con un estándar de comunicación celular, como 4G, 4G LTE (Long-Term Evolution), LTE Advanced, 5G o similar. En algunos ejemplos donde la interfaz de salida 108 comprende un transmisor inalámbrico, la interfaz de salida 108 y la interfaz de entrada 122 se pueden configurar para transferir datos, como datos de video codificados, de acuerdo con otros estándares inalámbricos, como una especificación IEEE 802.11, una especificación IEEE 802.15 (por ejemplo, ZigBee™), un estándar Bluetooth™ o similar. En algunos ejemplos, el dispositivo de origen 102 y/o el dispositivo de destino 116 pueden incluir dispositivos respectivos de sistema en un chip (SoC). Por ejemplo, el dispositivo de origen 102 puede incluir un dispositivo SoC para realizar la funcionalidad atribuida al codificador de video 200 y/o la interfaz de salida 108, y el dispositivo de destino 116 puede incluir un dispositivo SoC para realizar la funcionalidad atribuida al decodificador de video 300 y/o la interfaz de entrada 122.

[0034] Las técnicas de la presente divulgación se pueden aplicar a la codificación de video en apoyo de cualquiera de una variedad de aplicaciones multimedia, tales como transmisiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de video de flujo continuo por Internet, tales como transmisión dinámica adaptativa sobre HTTP (DASH), video digital que se codifica en un medio de almacenamiento de datos, decodificación de video digital almacenado en un medio de almacenamiento de datos u otras aplicaciones.

[0035] La interfaz de entrada 122 del dispositivo de destino 116 recibe un flujo de bits de video codificado del medio legible por computadora 110 (por ejemplo, dispositivo de almacenamiento 112, servidor de archivos 114 o similar). El medio legible por computadora 110 de flujo de bits de video codificado puede incluir información de señalización definida por el codificador de video 200, que también es utilizado por el decodificador de video 300, como elementos de sintaxis que tienen valores que describen características y/o el procesamiento de bloques de video u otras unidades codificadas (por ejemplo, segmentos, imágenes, grupos de imágenes, secuencias o similares). El dispositivo de visualización 118 muestra imágenes decodificadas de los datos de video decodificados a un usuario. El dispositivo de visualización 118 puede representar cualquiera de una variedad de dispositivos de visualización tales como un tubo de rayos catódicos (CRT), una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodo orgánico emisor de luz (OLED) u otro tipo de dispositivo de visualización.

[0036] Aunque no se muestra en la FIG. 1, en algunos ejemplos, tanto el codificador de

video 200 como el decodificador de video 300 pueden estar integrados con un codificador de audio y/o decodificador de audio, y pueden incluir unidades MUX-DEMUX apropiadas, u otro *hardware* y/o *software*, para manejar flujos multiplexados que incluyen ambos audio y video en un flujo de datos común. Si corresponde, las unidades MUX-DEMUX pueden cumplir con el protocolo multiplexor ITU H.223 u otros protocolos como el protocolo de datagramas de usuario (UDP).

[0037] Tanto el codificador de video 200 como el decodificador de video 300 pueden implementarse como cualquiera de una variedad de circuitos de codificador y/o decodificador adecuados, tales como uno o más microprocesadores, procesadores de señal digital (DSP), circuitos integrados específicos de aplicación (ASIC), matrices de puerta programables en campo (FPGA), lógica discreta, *software*, *hardware*, *firmware* o cualquier combinación de estos. Cuando las técnicas se implementan parcialmente en el *software*, un dispositivo puede almacenar instrucciones para el *software* en un medio adecuado, no transitorio, legible por computadora y ejecutar las instrucciones en *hardware* con el uso de uno o más procesadores para realizar las técnicas de la presente divulgación. Tanto el codificador de video 200 como el decodificador de video 300 pueden incluirse en uno o más codificadores o decodificadores, cualquiera de los cuales puede integrarse como parte de un codificador/decodificador combinado (CODEC) en un dispositivo respectivo. Un dispositivo que incluye un codificador de video 200 y/o un decodificador de video 300 puede comprender un circuito integrado, un microprocesador y/o un dispositivo de comunicación inalámbrico, tal como un teléfono celular.

[0038] El codificador de video 200 y el decodificador de video 300 pueden operar de acuerdo con un estándar de codificación de video, como ITU-T H.265, también conocido como Codificación de Video de Alta Eficiencia (HEVC) o extensiones de este, tales como las extensiones de codificación de vista múltiple y/o de video escalable. Alternativamente, el codificador de video 200 y el decodificador de video 300 pueden operar de acuerdo con otros estándares de propiedad o de la industria, como el Modelo de prueba de exploración conjunta (JEM) o ITU-T H.266, también denominado Codificación de video versátil (VVC). Un borrador reciente del estándar VVC se describe en Bross, et al. “*Versatile Video Coding (Draft 4)*”, Equipo conjunto de expertos en video (JVET) de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 13a reunión: Marrakech, MA, 9-18 de enero de 2019, JVET-M1001-v6 (en adelante, “Borrador 4 de VVC”). Las técnicas de esta divulgación, sin embargo, no se limitan a ningún estándar de codificación particular.

[0039] En general, el codificador de video 200 y el decodificador de video 300 pueden

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

[0040] Esta divulgación puede referirse generalmente a la codificación (por ejemplo, codificación y decodificación) de imágenes para incluir el proceso de codificación o decodificación de datos de la imagen. De manera similar, esta divulgación puede referirse a la codificación de bloques de una imagen para incluir el proceso de codificación o decodificación de datos para los bloques, por ejemplo, predicción y/o codificación residual. Un flujo de bits de video codificado generalmente incluye una serie de valores para elementos de sintaxis representativos de decisiones de codificación (por ejemplo, modos de codificación) y la división de imágenes en bloques. Por lo tanto, las referencias a la codificación de una imagen o un bloque generalmente deben entenderse como valores de codificación para elementos de sintaxis que forman la imagen o el bloque.

[0041] El HEVC define varios bloques, que incluyen unidades de codificación (CU), unidades de predicción (PU) y unidades de transformación (TU). De acuerdo con HEVC, un codificador de video (como el codificador de video 200) divide una unidad de árbol de codificación (CTU) en CU según una estructura *quadtree*. Es decir, el codificador de video divide las CTU y CU en cuatro cuadrados iguales y no superpuestos, y cada nodo del *quadtree* tiene cero o cuatro nodos secundarios. Los nodos sin nodos secundarios pueden denominarse “nodos hoja”, y las CU de dichos nodos hoja pueden incluir una o más PU y/o una o más TU. El codificador de video puede particionar más PU y TU. Por ejemplo, en HEVC, un *quadtree* residual (RQT) representa la partición de TU. En HEVC,

las PU representan datos de interpredicción, mientras que las TU representan datos residuales. Las CU que se intrapredicen incluyen información de intrapredicción, como una indicación de modo interno.

[0042] Como otro ejemplo, el codificador de video 200 y el decodificador de video 300 pueden configurarse para funcionar de acuerdo con JEM o VVC. Según JEM o VVC, un codificador de video (como el codificador de video 200) divide una imagen en una pluralidad de unidades de árbol de codificación (CTU). El codificador de video 200 puede dividir una CTU de acuerdo con una estructura de árbol, tal como una estructura de árbol binario *quadtree* (QTBT) o una estructura de árbol de tipo múltiple (MTT). La estructura QTBT elimina los conceptos de múltiples tipos de partición, como la separación entre CU, PU y TU de HEVC. Una estructura QTBT incluye dos niveles: un primer nivel particionado de acuerdo con la partición *quadtree* y un segundo nivel particionado de acuerdo con la partición de árbol binario. Un nodo raíz de la estructura QTBT corresponde a una CTU. Los nodos hoja de los árboles binarios corresponden a unidades de codificación (CU).

[0043] En una estructura de partición MTT, los bloques pueden dividirse con el uso de una partición *quadtree* (QT), una partición de árbol binario (BT) y uno o más tipos de particiones de árbol triple (TT). Una partición de árbol triple es una partición en la que un bloque se divide en tres sub-bloques. En algunos ejemplos, una partición de árbol triple divide un bloque en tres sub-bloques sin dividir el bloque original a través del centro. Los tipos de partición en MTT (por ejemplo, QT, BT y TT) pueden ser simétricos o asimétricos.

[0044] En algunos ejemplos, el codificador de video 200 y el decodificador de video 300 pueden usar una sola estructura QTBT o MTT para representar cada uno de los componentes de luminancia y crominancia, mientras que en otros ejemplos, el codificador de video 200 y el decodificador de video 300 pueden usar dos o más estructuras QTBT o MTT, como una estructura QTBT/MTT para el componente de luminancia y otra estructura QTBT/MTT para ambos componentes de crominancia (o dos estructuras QTBT/MTT para los componentes de crominancia respectivos).

[0045] El codificador de video 200 y el decodificador de video 300 pueden configurarse para usar el particionamiento *quadtree* por HEVC, particionamiento QTBT, particionamiento MTT u otras estructuras de particionamiento. Para fines de explicación, la descripción de las técnicas de la presente divulgación se presenta con respecto a la partición QTBT. Sin embargo, debe entenderse que las técnicas de la presente divulgación también pueden aplicarse a los codificadores de video configurados para

usar la partición *quadtree* u otros tipos de partición.

[0046] Esta divulgación puede usar “NxN” y “N por N” indistintamente para referirse a las dimensiones de muestra de un bloque (como una CU u otro bloque de video) en términos de dimensiones verticales y horizontales, por ejemplo, muestras de 16x16 o muestras de 16 por 16. En general, una CU de 16x16 tendrá 16 muestras en dirección vertical ($y = 16$) y 16 muestras en dirección horizontal ($x = 16$). Del mismo modo, una CU de NxN generalmente tiene N muestras en dirección vertical y N muestras en dirección horizontal, donde N representa un valor entero no negativo. Las muestras en una CU pueden organizarse en filas y columnas. Además, las CU no necesitan necesariamente tener el mismo número de muestras en la dirección horizontal que en la dirección vertical. Por ejemplo, las CU pueden comprender muestras de NxM, donde M no es necesariamente igual a N.

[0047] El codificador de video 200 codifica datos de video para CU que representan predicción y/o información residual, y otra información. La información de predicción indica cómo se debe predecir la CU para formar un bloque de predicción para la CU. La información residual generalmente representa diferencias muestra por muestra entre muestras de CU antes de la codificación y el bloque de predicción.

[0048] Para predecir una CU, el codificador de video 200 generalmente puede formar un bloque de predicción para la CU a través de la interpredicción o la intrapredicción. La interpredicción generalmente se refiere a la predicción de la CU a partir de datos de una imagen previamente codificada, mientras que la intrapredicción generalmente se refiere a la predicción de la CU a partir de datos previamente codificados de la misma imagen. Para realizar la interpredicción, el codificador de video 200 puede generar el bloque de predicción con el uso de uno o más vectores de movimiento. El codificador de video 200 generalmente puede realizar una búsqueda de movimiento para identificar un bloque de referencia que coincida estrechamente con la CU, por ejemplo, en términos de diferencias entre la CU y el bloque de referencia. El codificador de video 200 puede calcular una métrica de diferencia con el uso de una suma de diferencia absoluta (SAD), suma de diferencias cuadradas (SSD), diferencia absoluta media (MAD), diferencias cuadradas medias (MSD) u otros cálculos de diferencia similares para determinar si un bloque de referencia coincide estrechamente con la CU actual. En algunos ejemplos, el codificador de video 200 puede predecir la CU actual con el uso de predicción unidireccional o predicción bidireccional.

[0049] 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 afin, el codificador de video 200 puede determinar dos o más vectores de movimiento que representan movimiento no traslacional, tal como acercar o alejar, rotación, movimiento en perspectiva u otros tipos de movimiento irregular.

[0050] Para realizar la 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, incluidos varios modos direccionales, así como el modo plano y el modo DC. En general, el codificador de video 200 selecciona un modo de intrapredicción que describe las muestras vecinas a un bloque actual (por ejemplo, un bloque de una CU) desde el cual predecir muestras del bloque actual. Tales muestras generalmente pueden estar arriba, arriba y a la izquierda, o a la izquierda del bloque actual en la misma imagen que el bloque actual, suponiendo que el codificador de video 200 codifique CTU y CU en orden de exploración de trama (de izquierda a derecha, de arriba a abajo).

[0051] El codificador de video 200 codifica datos que representan el modo de predicción para un bloque actual. Por ejemplo, para los modos de interpredicción, el codificador de video 200 puede codificar datos que representan cuál de los diversos modos de interpredicción disponibles se usa, así como información de movimiento para el modo correspondiente. Para la interpredicción unidireccional o bidireccional, por ejemplo, el codificador de video 200 puede codificar vectores de movimiento con el uso de la predicción avanzada de vectores de movimiento (AMVP) o el modo de fusión. El codificador de video 200 puede usar modos similares para codificar vectores de movimiento para el modo de compensación de movimiento afin.

[0052] Después de la predicción, tal como la intrapredicción o la interpredicción de un bloque, el codificador de video 200 puede calcular datos residuales para el bloque. Los datos residuales, como un bloque residual, representan diferencias muestra por muestra entre el bloque y un bloque de predicción para el bloque, formado con el uso del modo de predicción correspondiente. El codificador de video 200 puede aplicar una o más transformadas al bloque residual, para producir datos transformados en un dominio de transformación en lugar del dominio de muestra. Por ejemplo, el codificador de video 200 puede aplicar una transformada discreta de coseno (DCT), una transformada entera, una transformada wavelet o una transformada conceptualmente similar a los datos de video residual. Además, el codificador de video 200 puede aplicar una transformada secundaria después de la primera transformación, tal como una transformada secundaria no separable dependiente del modo (MDNSST), una

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

[0053] Como se discutió con anterioridad, un codificador de video, tal como un codificador de video 200, puede aplicar varios tipos de transformaciones para transformar los datos residuales. La siguiente es una descripción general de las transformadas de seno y coseno discretas (DCT y DST). Además, se analiza brevemente el esquema de transformación utilizado en el estándar HEVC.

[0054] Transformadas discretas de seno y coseno.

[0055] La transformación indica el proceso de derivar una representación alternativa de la señal de entrada. Dado un vector de N puntos $\mathbf{x} = [x_0, x_1, \dots, x_{N-1}]^T$ y un conjunto de vectores dados $\{\phi_0, \phi_1, \dots, \phi_{M-1}\}$, $\mathbf{x}$ se puede aproximar o representar exactamente con el uso de una combinación lineal de $\phi_0, \phi_1, \dots, \phi_{M-1}$, que se puede formular de la siguiente manera,

$$\hat{\mathbf{x}} = \sum_{i=0}^{M-1} f_i \cdot \phi_i$$

[0056] donde $\hat{\mathbf{x}}$ puede ser una aproximación o equivalente de $\mathbf{x}$, el vector $\mathbf{f} = [f_0, f_1, \dots, f_{M-1}]$ se llama vector de coeficiente de transformación y $\{\phi_0, \phi_1, \dots, \phi_{M-1}\}$ son los vectores base de transformación.

[0057] En el escenario de la codificación de video, los coeficientes de transformación están aproximadamente no correlacionados y son escasos, es decir, la energía del vector de entrada $\mathbf{x}$ se compacta solo en unos pocos coeficientes de transformación, y los coeficientes de transformación mayoritarios restantes suelen estar cerca de 0.

[0058] Dados los datos de entrada específicos, la transformación óptima en términos de compactación de energía es la llamada transformada de Karhunen-Loeve (KLT), que utiliza los vectores propios de la matriz de covarianza de los datos de entrada como vectores base de la transformación. Por lo tanto, KLT es en realidad una transformación dependiente de datos y no tiene una formulación matemática general. Sin embargo, en ciertos supuestos, por ejemplo, los datos de entrada forman un proceso de Markov estacionario de primer orden, se ha demostrado en la literatura que la KLT correspondiente es en realidad un miembro de la familia sinusoidal de transformadas unitarias. La familia sinusoidal de transformadas unitarias indica transformaciones con el uso de vectores de base de transformación formulados de la siguiente manera:

$$\phi_m(k) = A \cdot e^{ik\theta} + B \cdot e^{-ik\theta}$$

[0059] donde e es la base del logaritmo natural aproximadamente igual a 2,71828, A, B

y θ son complejos en general y dependen del valor de m .

[0060] Las transformadas de ejemplo incluyen Fourier discreta, el coseno, el seno y KLT (para procesos de Markov estacionarios de primer orden) son miembros de esta familia sinusoidal de transformadas unitarias. Según S. A. Martucci, “*Symmetric convolution and the discrete sine and cosine transforms*”, IEEE Trans. Sig. Processing SP-42, 1038-1051 (1994), el conjunto completo de familias de transformada de coseno discreta (DCT) y transformada de seno discreta (DST) incluye un total de 16 transformaciones basadas en diferentes tipos, es decir, diferentes valores de A , B y θ , y una definición completa de los diferentes tipos de DCT y DST se dan a continuación.

[0061] Suponer que el vector de N puntos de entrada se denota como $\mathbf{x} = [x_0, x_1, \dots, x_{N-1}]^T$, y se transforma en otro vector de coeficiente de transformación de N puntos denotado como $\mathbf{y} = [y_0, y_1, \dots, y_{N-1}]^T$ al multiplicar una matriz, el proceso de la cual se puede ilustrar adicionalmente de acuerdo con una de las siguientes formulaciones de transformación, en donde k varía de 0 a $N-1$, inclusive:

[0062] DCT Tipo-I (DCT-1):

$$y_k = \sum_{n=0}^{N-1} \sqrt{\frac{2}{N-1}} \cos\left(\frac{\pi \cdot n \cdot k}{N-1}\right) \cdot w_0 \cdot w_1 \cdot x_n,$$

$$\text{donde } w_0 = \begin{cases} \frac{1}{\sqrt{2}}, & \text{si } n = 0 \text{ o } n = N-1 \\ 1, & \text{si no} \end{cases}, w_1 = \begin{cases} \frac{1}{\sqrt{2}}, & \text{si } k = 0 \text{ o } k = N-1 \\ 1, & \text{si no} \end{cases}$$

[0063] DCT Tipo-II (DCT-2):

$$y_k = \sum_{n=0}^{N-1} \sqrt{\frac{2}{N}} \cos\left(\frac{\pi \cdot (n+0.5) \cdot k}{N-1}\right) \cdot w_0 \cdot x_n,$$

$$\text{donde } w_0 = \begin{cases} \frac{1}{\sqrt{2}}, & \text{si } k = 0 \\ 1, & \text{si no} \end{cases}$$

[0064] DCT Tipo-III (DCT-3):

$$y_k = \sum_{n=0}^{N-1} \sqrt{\frac{2}{N}} \cos\left(\frac{\pi \cdot n \cdot (k+0.5)}{N}\right) \cdot w_0 \cdot x_n,$$

$$\text{donde } w_0 = \begin{cases} \frac{1}{\sqrt{2}}, & \text{si } n = 0 \\ 1, & \text{si no} \end{cases}$$

[0065] DCT Tipo-IV (DCT-4):

$$y_k = \sum_{n=0}^{N-1} \sqrt{\frac{2}{N}} \cos\left(\frac{\pi \cdot (n+0.5) \cdot (k+0.5)}{N}\right) \cdot x_n,$$

[0066] DCT Tipo-V (DCT-5):

$$y_k = \sum_{n=0}^{N-1} \sqrt{\frac{2}{N-0.5}} \cos\left(\frac{\pi \cdot n \cdot k}{N-0.5}\right) \cdot w_0 \cdot w_1 \cdot x_n,$$

$$\text{donde } w_0 = \begin{cases} \frac{1}{\sqrt{2}}, & \text{si } n = 0 \\ 1, & \text{si no} \end{cases}, w_1 = \begin{cases} \frac{1}{\sqrt{2}}, & \text{si } k = 0 \\ 1, & \text{si no} \end{cases}$$

[0067] DCT Tipo-VI (DCT-6):

$$y_k = \sum_{n=0}^{N-1} \sqrt{\frac{2}{N-0.5}} \cos \left(\frac{\pi \cdot (n+0.5) \cdot k}{N-0.5} \right) \cdot w_0 \cdot w_1 \cdot x_n,$$

$$\text{donde } w_0 = \begin{cases} \frac{1}{\sqrt{2}}, & \text{si } n = N - 1 \\ 1, & \text{si no} \end{cases}, w_1 = \begin{cases} \frac{1}{\sqrt{2}}, & \text{si } k = 0 \\ 1, & \text{si no} \end{cases}$$

[0068] DCT Tipo-VII (DCT-7):

$$y_k = \sum_{n=0}^{N-1} \sqrt{\frac{2}{N-0.5}} \cos \left(\frac{\pi \cdot n \cdot (k+0.5)}{N-0.5} \right) \cdot w_0 \cdot w_1 \cdot x_n,$$

$$\text{donde } w_0 = \begin{cases} \frac{1}{\sqrt{2}}, & \text{si } n = 0 \\ 1, & \text{si no} \end{cases}, w_1 = \begin{cases} \frac{1}{\sqrt{2}}, & \text{si } k = N - 1 \\ 1, & \text{si no} \end{cases}$$

[0069] DCT Tipo-VIII (DCT-8):

$$y_k = \sum_{n=0}^{N-1} \sqrt{\frac{2}{N+0.5}} \cos \left(\frac{\pi \cdot (n+0.5) \cdot (k+0.5)}{N+0.5} \right) \cdot x_n,$$

[0070] DST Tipo-I (DST-1):

$$y_k = \sum_{n=0}^{N-1} \sqrt{\frac{2}{N+1}} \sin \left(\frac{\pi \cdot (n+1) \cdot (k+1)}{N+1} \right) \cdot x_n,$$

[0071] DST Tipo-II (DST-2):

$$y_k = \sum_{n=0}^{N-1} \sqrt{\frac{2}{N}} \sin \left(\frac{\pi \cdot (n+0.5) \cdot (k+1)}{N} \right) \cdot w_0 \cdot x_n,$$

$$\text{donde } w_0 = \begin{cases} \frac{1}{\sqrt{2}}, & \text{si } k = N - 1 \\ 1, & \text{si no} \end{cases}$$

[0072] DST Tipo-III (DST-3):

$$y_k = \sum_{n=0}^{N-1} \sqrt{\frac{2}{N}} \sin \left(\frac{\pi \cdot (n+1) \cdot (k+0.5)}{N} \right) \cdot w_0 \cdot x_n,$$

$$\text{donde } w_0 = \begin{cases} \frac{1}{\sqrt{2}}, & \text{si } n = N - 1 \\ 1, & \text{si no} \end{cases}$$

[0073] DST Tipo-IV (DST-4):

$$y_k = \sum_{n=0}^{N-1} \sqrt{\frac{2}{N}} \sin \left(\frac{\pi \cdot (n+0.5) \cdot (k+0.5)}{N} \right) \cdot x_n,$$

[0074] DST Tipo-V (DST-5):

$$y_k = \sum_{n=0}^{N-1} \sqrt{\frac{2}{N+0.5}} \sin \left(\frac{\pi \cdot (n+1) \cdot (k+1)}{N+0.5} \right) \cdot x_n,$$

[0075] DST Tipo-VI (DST-6):

$$y_k = \sum_{n=0}^{N-1} \sqrt{\frac{2}{N+0.5}} \sin \left(\frac{\pi \cdot (n+0.5) \cdot (k+1)}{N+0.5} \right) \cdot x_n,$$

[0076] DST Tipo-VII (DST-7):

$$y_k = \sum_{n=0}^{N-1} \sqrt{\frac{2}{N+0.5}} \sin \left(\frac{\pi \cdot (n+1) \cdot (k+0.5)}{N+0.5} \right) \cdot x_n,$$

[0077] DST Tipo-VIII (DST-8):

$$y_k = \sum_{n=0}^{N-1} \sqrt{\frac{2}{N-0.5}} \cos \left(\frac{\pi \cdot (n+0.5) \cdot (k+0.5)}{N-0.5} \right) \cdot w_0 \cdot w_1 \cdot x_n,$$

$$\text{donde } w_0 = \begin{cases} \frac{1}{\sqrt{2}}, & \text{si } n = N - 1 \\ 1, & \text{si no} \end{cases}, w_1 = \begin{cases} \frac{1}{\sqrt{2}}, & \text{si } k = N - 1 \\ 1, & \text{si no} \end{cases}$$

[0078] El tipo de transformación se especifica mediante la formulación matemática de la función de base de transformación, por ejemplo, DST-VII de 4 puntos y DST-VII de 8 puntos tienen el mismo tipo de transformación, independientemente del valor de N.

[0079] Sin pérdida de generalidad, todos los tipos de transformación anteriores se pueden representar mediante la siguiente formulación generalizada:

$$y_m = \sum_{n=0}^{N-1} T_{m,n} \cdot x_n,$$

[0080] donde T es la matriz de transformación especificada por la definición de una determinada transformada, por ejemplo, DCT Tipo-I ~ DCT Tipo-VIII, o DST Tipo-I ~ DST Tipo-VIII, y los vectores de fila de T, por ejemplo, $[T_{i,0}, T_{i,1}, T_{i,2}, \dots, T_{i,N-1}]$ son los iésimos vectores base de la transformación. Una transformada aplicada en el vector de entrada de N puntos se llama transformada de N puntos.

[0081] También se observa que, las formulaciones de transformación anteriores, que se aplican en los datos de entrada 1-D x, se pueden representar en forma de multiplicación de matrices como se muestra a continuación

$$y = T \cdot x$$

[0082] donde T indica la matriz de transformación, x indica el vector de datos de entrada e y indica el vector de coeficientes de transformación de salida.

[0083] Transformación para datos de entrada bidimensionales (2-D).

[0084] Las transformaciones introducidas en la sección anterior se aplican a datos de entrada 1-D, y las transformaciones también se pueden extender para fuentes de datos de entrada 2-D. Suponiendo que X es una matriz de datos de entrada MxN, los métodos típicos de aplicar la transformación en datos de entrada 2-D incluyen las transformadas 2-D separables y no separables.

[0085] Una transformada 2-D separable aplica transformadas 1-D para los vectores horizontal y vertical de X secuencialmente, formuladas como se muestra a continuación:

$$Y = C \cdot X \cdot R^T$$

[0086] donde C y R indican las matrices de transformación MxM y NxN dadas, respectivamente. De la formulación, se puede ver que C aplica transformadas 1-D para

los vectores columna de X , mientras que R aplica transformadas 1-D para los vectores fila de X . En la última parte de este documento, por simplicidad, se denotan C y R como transformadas izquierda (vertical) y derecha (horizontal) y ambas forman un par de transformadas. Hay casos en los que C es igual a R y es una matriz ortogonal. En tal caso, la transformada 2-D separable se determina mediante una sola matriz de transformación.

[0087] Una transformada 2-D no separable primero reorganizó todos los elementos de X en un solo vector, a saber, X' , mediante el siguiente mapeo matemático como ejemplo:

$$X'_{(i \cdot N + j)} = X_{i,j}$$

[0088] Entonces se aplica una transformada 1-D T' para X' como se muestra a continuación:

$$Y = T' \cdot X$$

[0089] donde T' es una matriz de transformación $(M \cdot N) \times (M \cdot N)$.

[0090] En la codificación de video, se pueden aplicar transformadas 2-D separables ya que pueden utilizar mucho menos recuentos de operaciones (suma, multiplicación) en comparación con la transformada 1-D.

[0091] En códecs de video convencionales, tales como H.264/AVC, siempre se aplica una aproximación entera de la Transformada de Coseno Discreta (DCT) Tipo II de 4 puntos y 8 puntos para el residuo de intrapredicción e interpredicción. Para adaptarse mejor a las diversas estadísticas de muestras residuales, en los códecs de video más nuevos se utilizan tipos de transformaciones más flexibles que no sean DCT Tipo II. Por ejemplo, en HEVC, se utiliza una aproximación entera de la Transformada sinusoidal discreta (DST) Tipo VII de 4 puntos para el residuo de intrapredicción, que está teóricamente probado y validado experimentalmente que DST Tipo-VII es más eficiente que DCT Tipo-II para vectores residuales generados a lo largo de las direcciones de intrapredicción, por ejemplo, DST Tipo-VII es más eficiente que DCT Tipo-II para vectores residuales de fila generados por la dirección de intrapredicción horizontal. En HEVC, una aproximación entera de DST Tipo-VII de 4 puntos se aplica solo para bloques residuales de intrapredicción de luma de 4x4. La DST-VII de 4 puntos utilizado en HEVC se muestra a continuación,

[0092] 4x4 DST-VII:

{29, 55, 74, 84}

{74, 74, 0, -74}

{84, -29, -74, 55}

{55, -84, 74, -29}

[0093] En HEVC, para bloques residuales que no son bloques residuales de intrapredicción de luma de 4x4, también se aplican aproximaciones enteras de la DCT Tipo II de 4 puntos, 8 puntos, 16 puntos y 32 puntos, como se muestra a continuación:

[0094] DCT-II de 4 puntos:

{64, 64, 64, 64}

{83, 36,-36,-83}

{64,-64,-64, 64}

{36,-83, 83,-36}

[0095] DCT-II de 8 puntos:

{64, 64, 64, 64, 64, 64, 64, 64}

{89, 75, 50, 18,-18,-50,-75,-89}

{83, 36,-36,-83,-83,-36, 36, 83}

{75,-18,-89,-50, 50, 89, 18,-75}

{64,-64,-64, 64, 64,-64,-64, 64}

{50,-89, 18, 75,-75,-18, 89,-50}

{36,-83, 83,-36,-36, 83,-83, 36}

{18,-50, 75,-89, 89,-75, 50,-18}

[0096] DCT-II de 16 puntos:

{64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64}

{90, 87, 80, 70, 57, 43, 25, 9, -9,-25,-43,-57,-70,-80,-87,-90}

{89, 75, 50, 18,-18,-50,-75,-89,-89,-75,-50,-18, 18, 50, 75, 89}

{87, 57, 9,-43,-80,-90,-70,-25, 25, 70, 90, 80, 43, -9,-57,-87}

{83, 36,-36,-83,-83,-36, 36, 83, 83, 36,-36,-83,-83,-36, 36, 83}

{80, 9,-70,-87,-25, 57, 90, 43,-43,-90,-57, 25, 87, 70, -9,-80}

{75,-18,-89,-50, 50, 89, 18,-75,-75, 18, 89, 50,-50,-89,-18, 75}

{70,-43,-87, 9, 90, 25,-80,-57, 57, 80,-25,-90, -9, 87, 43,-70}

{64,-64,-64, 64, 64,-64,-64, 64, 64,-64,-64, 64, 64,-64,-64, 64}

{57,-80,-25, 90, -9,-87, 43, 70,-70,-43, 87, 9,-90, 25, 80,-57}

{50,-89, 18, 75,-75,-18, 89,-50,-50, 89,-18,-75, 75, 18,-89, 50}

{43,-90, 57, 25,-87, 70, 9,-80, 80, -9,-70, 87,-25,-57, 90,-43}

{36,-83, 83,-36,-36, 83,-83, 36, 36,-83, 83,-36,-36, 83,-83, 36}

{25,-70, 90,-80, 43, 9,-57, 87,-87, 57, -9,-43, 80,-90, 70,-25}

{18,-50, 75,-89, 89,-75, 50,-18,-18, 50,-75, 89,-89, 75,-50, 18}

{9, -25, 43,-57, 70,-80, 87,-90, 90,-87, 80,-70, 57,-43, 25, -9}

[0097] DCT-II de 32 puntos:

90,13,88,-31,-82,46,73,-61}
 {57,-80,-25,90,-9,-87,43,70,-70,-43,87,9,-90,25,80,-57,-57,80,25,-90,9,87,-43,-
 70,70,43,-87,-9,90,-25,-80,57}
 {54,-85,-4,88,-46,-61,82,13,-90,38,67,-78,-22,90,-31,-73,73,31,-90,22,78,-67,-38,90,-
 13,-82,61,46,-88,4,85,-54}
 {50,-89,18,75,-75,-18,89,-50,-50,89,-18,-75,75,18,-89,50,50,-89,18,75,-75,-18,89,-50,-
 50,89,-18,-75,75,18,-89,50}
 {46,-90,38,54,-90,31,61,-88,22,67,-85,13,73,-82,4,78,-78,-4,82,-73,-13,85,-67,-22,88,-
 61,-31,90,-54,-38,90,-46}
 {43,-90,57,25,-87,70,9,-80,80,-9,-70,87,-25,-57,90,-43,-43,90,-57,-25,87,-70,-9,80,-
 80,9,70,-87,25,57,-90,43}
 {38,-88,73,-4,-67,90,-46,-31,85,-78,13,61,-90,54,22,-82,82,-22,-54,90,-61,-13,78,-
 85,31,46,-90,67,4,-73,88,-38}
 {36,-83,83,-36,-36,83,-83,36,36,-83,83,-36,-36,83,-83,36,36,-83,83,-36,-36,83,-
 83,36,36,-83,83,-36,-36,83,-83,36}
 {31,-78,90,-61,4,54,-88,82,-38,-22,73,-90,67,-13,-46,85,-85,46,13,-67,90,-73,22,38,-
 82,88,-54,-4,61,-90,78,-31}
 {25,-70,90,-80,43,9,-57,87,-87,57,-9,-43,80,-90,70,-25,-25,70,-90,80,-43,-9,57,-87,87,-
 57,9,43,-80,90,-70,25}
 {22,-61,85,-90,73,-38,-4,46,-78,90,-82,54,-13,-31,67,-88,88,-67,31,13,-54,82,-90,78,-
 46,4,38,-73,90,-85,61,-22}
 {18,-50,75,-89,89,-75,50,-18,-18,50,-75,89,-89,75,-50,18,18,-50,75,-89,89,-75,50,-18,-
 18,50,-75,89,-89,75,-50,18}
 {13,-38,61,-78,88,-90,85,-73,54,-31,4,22,-46,67,-82,90,-90,82,-67,46,-22,-4,31,-54,73,-
 85,90,-88,78,-61,38,-13}
 {9,-25,43,-57,70,-80,87,-90,90,-87,80,-70,57,-43,25,-9,-9,25,-43,57,-70,80,-87,90,-
 90,87,-80,70,-57,43,-25,9}
 {4,-13,22,-31,38,-46,54,-61,67,-73,78,-82,85,-88,90,-90,90,-90,88,-85,82,-78,73,-
 67,61,-54,46,-38,31,-22,13,-4}

[0098] Esquema de transformación basado en *quadtree* residual en HEVC.

[0099] Para adaptar las diversas características de los bloques residuales, se aplica una estructura de codificación de transformada que utiliza el *quadtree* residual (RQT) en HEVC, que se describe brevemente en <http://www.hhi.fraunhofer.de/fields-of-competence/image-processing/research-groups/image-video-coding/hevc-high-efficiency-video-coding/transform-coding-using-the-residual-quadtree-rqt.html>. En RQT,

cada imagen se divide en unidades de árbol de codificación (CTU), que se codifican en orden de exploración de trama para un mosaico o segmento específico. Una CTU es un bloque cuadrado y representa la raíz de un *quadtree*, es decir, el árbol de codificación. El tamaño de la CTU puede oscilar entre 8×8 y 64×64 muestras de luma, pero normalmente se utiliza 64×64 . Cada CTU se puede dividir en bloques cuadrados más pequeños llamados unidades de codificación (CU). Después de que la CTU se divide recursivamente en CU, cada CU se divide en unidades de predicción (PU) y unidades de transformación (TU).

[0100] La partición de una CU en TU se puede llevar a cabo de forma recursiva en base a un enfoque de *quadtree*, por lo tanto, la señal residual de cada CU está codificada por una estructura de árbol, a saber, el *quadtree* residual (RQT). El RQT permite tamaños de TU desde 4×4 hasta 32×32 muestras de luma. La FIG. 2C muestra un ejemplo en donde una CU incluye 10 TU, etiquetadas con las letras de la a a la j, y la partición de bloque correspondiente. Cada nodo del RQT es en realidad una unidad de transformación (TU). Las TU individuales se pueden procesar en el orden de recorrido de árbol en profundidad primero, que se ilustra en la figura como orden alfabético, que sigue a una exploración Z recursiva con recorrido en profundidad primero.

[0101] El enfoque de *quadtree* permite la adaptación de la transformada a las características de frecuencia espacial variables de la señal residual. Normalmente, los tamaños de bloque de transformación más grandes, que tienen un mayor soporte espacial, proporcionan una mejor resolución de frecuencia. Sin embargo, los tamaños de bloques de transformación más pequeños, que tienen un soporte espacial más pequeño, pueden proporcionar una mejor resolución espacial. La compensación entre las dos, resolución espacial y de frecuencia, puede elegirse mediante la decisión del modo de codificador, por ejemplo, en función de la técnica de optimización de la distorsión de la velocidad. El codificador de video puede realizar una técnica de optimización de distorsión de velocidad para calcular una suma ponderada de bits de codificación y distorsión de reconstrucción, es decir, el costo de distorsión de velocidad, para cada modo de codificación (por ejemplo, una estructura de división RQT específica) y seleccionar el modo de codificación con el menor costo de distorsión de velocidad como el mejor modo.

[0102] Se pueden definir tres parámetros en el RQT: la profundidad máxima del árbol, el tamaño de transformación mínimo permitido y el tamaño de transformación máximo permitido. Los tamaños de transformación mínimo y máximo pueden variar dentro del rango de 4×4 a 32×32 muestras, que corresponden a las transformaciones de bloque

soportadas mencionadas en el párrafo anterior. La profundidad máxima permitida del RQT restringe el número de TU. Una profundidad máxima igual a cero significa que un CB no se puede dividir más si cada TB incluido alcanza el tamaño de transformación máximo permitido, por ejemplo, 32×32 .

[0103] Todos estos parámetros interactúan e influyen en la estructura de RQT. Considerar un caso, en donde el tamaño del CB raíz es 64×64 , la profundidad máxima es igual a cero y el tamaño máximo de transformación es igual a 32×32 . En este caso, el CB debe dividirse al menos una vez, ya que de lo contrario conduciría a un TB de 64×64 , que no está permitido. Los parámetros de RQT, es decir, la profundidad máxima de RQT, el tamaño de transformación mínimo y máximo, se transmiten en el flujo de bits en el nivel de conjunto de parámetros de secuencia. Con respecto a la profundidad de RQT, se pueden especificar y señalar diferentes valores para las CU intracodificadas e intercodificadas.

[0104] La transformación *quadtree* se aplica para los bloques residuales intra e inter. Normalmente, la transformada DCT-II del mismo tamaño de la partición *quadtree* residual actual se aplica para un bloque residual. Sin embargo, si el bloque *quadtree* residual actual es de 4×4 y se genera mediante intrapredicción, se aplica la transformada DST-VII de 4×4 anterior.

[0105] En HEVC, las transformaciones de mayor tamaño, por ejemplo, la transformación de 64×64 no se adoptan principalmente debido a su beneficio limitado considerando y complejidad relativamente alta para videos de resolución relativamente más pequeña.

[0106] Como se indicó con anterioridad, después de cualquier transformación para producir coeficientes de transformación, el codificador de video 200 puede realizar la cuantificación de los coeficientes de transformación. La cuantificación generalmente se refiere a un proceso en donde los coeficientes de transformación se cuantifican para posiblemente reducir la cantidad de datos utilizados para representar los coeficientes, proporcionando así una compresión adicional. Al realizar el proceso de cuantificación, el codificador de video 200 puede reducir la profundidad de bits asociada con algunos o todos los coeficientes. Por ejemplo, el codificador de video 200 puede redondear un valor de n bits a un valor de m bits durante la cuantificación, donde n es mayor que m . En algunos ejemplos, para realizar la cuantificación, el codificador de video 200 puede realizar un desplazamiento a la derecha bit a bit del valor a cuantificar.

[0107] Después de la cuantificación, el codificador de video 200 puede explorar los coeficientes de transformación, produciendo un vector unidimensional a partir de la matriz bidimensional que incluye los coeficientes de transformación cuantificados. La

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

[0108] Para realizar CABAC, el codificador de video 200 puede asignar un contexto dentro de un modelo de contexto a un símbolo a transmitir. El contexto puede relacionarse con, por ejemplo, si los valores vecinos del símbolo tienen valor cero o no. La determinación de probabilidad puede basarse en un contexto asignado al símbolo.

[0109] El codificador de video 200 puede generar además datos de sintaxis, tales como datos de sintaxis basados en bloques, datos de sintaxis basados en imágenes y datos de sintaxis basados en secuencias, para el decodificador de video 300, por ejemplo, en un encabezado de imagen, un encabezado de bloque, un encabezado de segmento u otros datos de sintaxis, como un conjunto de parámetros de secuencia (SPS), un conjunto de parámetros de imagen (PPS) o un conjunto de parámetros de video (VPS). El decodificador de video 300 también puede decodificar dichos datos de sintaxis para determinar cómo decodificar los datos de video correspondientes.

[0110] 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) e información de predicción y/o residual para los bloques. Finalmente, el decodificador de video 300 puede recibir el flujo de bits y decodificar los datos de video codificados.

[0111] En general, el decodificador de video 300 realiza un proceso recíproco al realizado por el codificador de video 200 para decodificar los datos de video codificados del flujo de bits. Por ejemplo, el decodificador de video 300 puede decodificar valores

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

[0112] La información residual puede representarse, por ejemplo, por coeficientes de transformación cuantificados. El decodificador de video 300 puede cuantificar y transformar de forma inversa los coeficientes de transformación cuantificados de un bloque para reproducir un bloque residual para el bloque. El decodificador de video 300 usa un modo de predicción señalado (intrapredicción o interpredicción) e información de predicción relacionada (por ejemplo, información de movimiento para la interpredicción) para formar un bloque de predicción para el bloque. El decodificador de video 300 puede combinar entonces el bloque de predicción y el bloque residual (muestra por muestra) para reproducir el bloque original. El decodificador de video 300 puede realizar un procesamiento adicional, como realizar un proceso de desbloqueo para reducir los artefactos visuales a lo largo de los límites del bloque.

[0113] De acuerdo con las técnicas de esta divulgación, un codificador de video (es decir, un codificador de video 200 y/o un decodificador de video 300) puede derivar, para un bloque de coeficiente actual de un bloque de video, un tipo de transformación a partir de una pluralidad de tipos de transformación. El codificador de video puede transformar, con el uso del tipo de transformación seleccionado, el bloque de transformación actual (por ejemplo, bloque de coeficientes) para obtener un bloque de datos residuales reconstruidos para el bloque de video; y reconstruir, en función de los datos residuales reconstruidos para el bloque de video, el bloque de video.

[0114] El codificador de video puede inferir el tipo de transformación en función de factores distintos de una señalización expresa del tipo de transformación. Como tal, el codificador de video puede omitir la codificación de un elemento de sintaxis que identifica expresamente el tipo de transformación para un bloque actual. Algunos ejemplos de factores de los cuales el codificador de video puede inferir el tipo de transformación incluyen un tamaño del bloque actual (por ejemplo, una altura y/o un ancho del bloque actual), si el bloque actual está particionado mediante particiones intrasub-bloques (ISP) y un modo intra del bloque actual. El codificador de video puede inferir el tipo de transformación en función de cualquier combinación de factores. Por

ejemplo, el codificador de video puede inferir el tipo de transformación para un bloque de transformación actual de un bloque de video actual en función de un tamaño del bloque de transformación actual y si el bloque de video actual está particionado mediante ISP. En al menos algunos de tales ejemplos, el codificador de video puede inferir el tipo de transformación para el bloque de transformación actual independientemente del modo de intrapredicción utilizado para predecir el bloque de video actual.

[0115] El codificador de video puede seleccionar el tipo de transformación de una pluralidad de tipos de transformación que incluye una o más transformadas de coseno discretas (DCT) y una o más transformadas de seno discretas (DST). Como se analiza con más detalle a continuación, la DCT o las DCT pueden incluir uno o más de una DCT-1, una DCT-2, una DCT-3, una DCT-4, una DCT-5, una DCT-6, una DCT-7 y una DCT-8, y/o la DST o las DST pueden incluir una o más de una DST-1, una DST-2, una DST-3, una DST-4, una DST-5, una DST-6, una DST-7 y una DST-8.

[0116] Como se discutió con anterioridad, el codificador de video puede inferir el tipo de transformación para un bloque de transformación actual en función de un tamaño del bloque de transformación actual. Por ejemplo, el codificador de video puede seleccionar un primer tipo de transformación para el bloque de transformación actual en respuesta a determinar que el tamaño del bloque de transformación actual satisface un umbral de tamaño y seleccionar un segundo tipo de transformación para el bloque de transformación actual en respuesta a determinar que el tamaño del bloque de transformación actual no satisface el umbral de tamaño. En algunos ejemplos, el codificador de video puede determinar si el tamaño del bloque de transformación actual satisface el umbral de tamaño al comparar el tamaño del bloque de transformación actual con un único valor de umbral. En otros ejemplos, el codificador de video puede determinar si el tamaño del bloque de transformación actual satisface el umbral de tamaño al determinar si el tamaño del bloque de transformación actual es mayor que un límite inferior (por ejemplo, 2, 4, 6) y menor que un límite superior (por ejemplo, 8, 16, 32). Si el tamaño del bloque de transformación actual es mayor que el límite inferior y menor que el límite superior, el codificador de video puede determinar que el tamaño del bloque de transformación actual satisface el umbral de tamaño. De manera similar, si el tamaño del bloque de transformación actual es menor que el límite inferior o mayor que el límite superior, el codificador de video puede determinar que el tamaño del bloque de transformación actual no satisface el umbral de tamaño.

[0117] Cuando el bloque de video actual es una unidad de codificación (CU), la CU

puede dividirse en una pluralidad de subparticiones mediante ISP. Cada una de las subparticiones puede tener un bloque de transformación asociado. Como tal, cuando la CU se particiona mediante ISP, se pueden asociar una pluralidad de bloques de transformación con la CU. Por ejemplo, una CU de 16x16 puede dividirse verticalmente en cuatro particiones de tamaño 4x16, cada una de las cuales está asociada con un bloque de transformación de tamaño 4x16.

[0118] Como se discutió con anterioridad, el codificador de video puede inferir el tipo de transformación para un bloque de transformación actual de un bloque de video actual en función de si el bloque de video actual está particionado mediante ISP y basado en un tamaño del bloque de transformación actual. Como ejemplo, en respuesta a la determinación de que el tamaño del bloque de transformación actual satisface un umbral de tamaño y que el bloque de video actual está particionado mediante ISP, el codificador de video puede seleccionar una DST particular de una o más DST (por ejemplo, DST-7) como el tipo de transformación para el bloque de transformación actual. Como otro ejemplo, en respuesta a determinar que el tamaño del bloque de transformación actual no satisface el umbral de tamaño y que el bloque de video actual está particionado mediante ISP, el codificador de video puede seleccionar una DCT particular de una o más DCT (por ejemplo, DCT-2) como el tipo de transformación para el bloque de transformación actual. En cualquiera de los ejemplos mencionados con anterioridad, el codificador de video puede seleccionar el tipo de transformación que comprende seleccionar el tipo de transformación independientemente de un modo de intrapredicción usado para predecir el bloque de video actual (por ejemplo, independientemente del modo angular, DC o plano usado para intraprededir el video actual).

[0119] En algunos ejemplos, el codificador de video siempre puede realizar la inferencia del tipo de transformación. En otros ejemplos, el codificador de video puede realizar la inferencia del tipo de transformación en ciertas condiciones. Por ejemplo, el codificador de video puede inferir el tipo de transformación para el bloque de transformación actual en respuesta a determinar que la selección de transformación múltiple (MTS) está habilitada para el bloque de video actual. El codificador de video puede, en algunos ejemplos, determinar si MTS está habilitado para el bloque de video actual en función de los valores de uno o más elementos de sintaxis (por ejemplo, `sps_explicit_mts_intra_enabled_flag`).

[0120] El codificador de video puede, en algunos ejemplos, inferir un tipo de transformación para realizar transformaciones horizontales (es decir, un tipo de transformación para uso horizontal) e inferir un tipo de transformación para realizar

transformaciones verticales (es decir, un tipo de transformación para uso vertical). El codificador de video puede inferir los tipos de transformación para uso horizontal y vertical mediante un algoritmo común. Por ejemplo, el codificador de video puede inferir el tipo de transformación para uso horizontal en función de si el ancho de un bloque de transformación actual satisface un umbral de tamaño de ancho y si un bloque de video actual que incluye el bloque de transformación actual está particionado mediante ISP e inferir el tipo de transformación para uso vertical en función de si una altura del bloque de transformación actual satisface un umbral de tamaño de altura y si el bloque de video actual que incluye el bloque de transformación actual está particionado mediante ISP. En algunos ejemplos, el codificador de video puede usar el mismo umbral de tamaño para inferencias de tipo de transformación horizontal y vertical. Por ejemplo, cuando los umbrales de tamaño incluyen un límite superior y un límite inferior, los límites superior e inferior del umbral de tamaño de ancho pueden ser iguales a los límites superior e inferior del umbral de tamaño de altura. Como ejemplo específico, el límite inferior de los umbrales de ancho y alto puede ser 4 y el límite superior de los umbrales de ancho y alto puede ser 16.

[0121] En algunos ejemplos, para derivar (es decir, inferir) el tipo de transformación para el bloque de coeficiente actual, el codificador de video puede seleccionar la transformada DST-7 para transformar cualquier fila o columna con un número de muestras (por ejemplo, muestras de luma) menor o igual al umbral (por ejemplo, 8, 16, 32) y seleccione la transformada DCT-2 para transformar cualquier fila o columna con un número de muestras mayor que el umbral.

[0122] En relación con el Borrador 4 de VVC (por ejemplo, JVET-M1001), se puede lograr un ejemplo del cambio propuesto al reemplazar la Tabla 8-15 con lo siguiente:

$$\text{trTypeHor} = (\text{nTbW} \geq 2 \ \&\& \ \text{nTbW} \leq 16) ? 1 : 0$$

$$\text{trTypeVer} = (\text{nTbH} \geq 2 \ \&\& \ \text{nTbH} \leq 16) ? 1 : 0$$

donde “0” y “1” denotan DCT-2 y DST-7, respectivamente.

[0123] Se puede prohibir que los bloques particionados mediante ISP tengan filas/columnas con solo dos muestras. Como tal, esta divulgación propone una DST-7 de 2 puntos. Las entradas de la matriz DST-7 de 2 puntos pueden ser las siguientes (que solo introduce 4 bytes de memoria adicional):

$$\{ 48 \ 77 \}$$

$$\{ 77 \ -48 \}$$

[0124] Alternativamente, se puede lograr un ejemplo del cambio propuesto al modificar el Borrador 4 de VVC de la siguiente manera:

$$\text{trTypeHor} = (\text{nTbW} \geq 4 \ \&\& \ \text{nTbW} \leq 16 \ \&\& \ \underline{\text{nTbW}} \leq \underline{\text{nTbH}}) ? 1 : 0 \quad (8-1029)$$

$$\text{trTypeVer} = (\text{nTbH} \geq 4 \ \&\& \ \text{nTbH} \leq 16 \ \&\& \ \underline{\text{nTbH}} \leq \underline{\text{nTbW}}) ? 1 : 0 \quad (8-1030)$$

donde “0” y “1” denotan DCT-2 y DST-7 respectivamente y los cambios (es decir, porciones eliminadas) están subrayados y en cursiva.

[0125] Esta divulgación generalmente puede referirse a “señalizar” cierta información, tal como elementos de sintaxis. El término “señalización” puede referirse generalmente a la comunicación de valores de elementos de sintaxis y/u otros datos utilizados para decodificar datos de video codificados. Es decir, el codificador de video 200 puede señalar valores para elementos de sintaxis en el flujo de bits. En general, la señalización se refiere a generar un valor en el flujo de bits. Como se indicó con anterioridad, el dispositivo de origen 102 puede transportar el flujo de bits al dispositivo de destino 116 sustancialmente en tiempo real, o no en tiempo real, tal como podría ocurrir al almacenar elementos de sintaxis en el dispositivo de almacenamiento 112 para su posterior recuperación por el dispositivo de destino 116.

[0126] Las FIG. 2A y 2B son diagramas conceptuales que ilustran un ejemplo de estructura de árbol binario *quadtree* (QTBT) 130, y una unidad de árbol de codificación correspondiente (CTU) 132. Las líneas continuas representan la división de *quadtree* y las líneas de puntos indican la división de árbol binario. En cada nodo de división (es decir, no hoja) del árbol binario, se señala un indicador para indicar qué tipo de división (es decir, horizontal o vertical) se utiliza, donde 0 indica división horizontal y 1 indica división vertical en este ejemplo. Para la división de *quadtree*, no es necesario indicar el tipo de división, ya que los nodos *quadtree* dividen un bloque horizontal y verticalmente en 4 sub-bloques con el mismo tamaño. En consecuencia, el codificador de video 200 puede codificar, y el decodificador de video 300 puede decodificar elementos de sintaxis (como la información de división) para un nivel de árbol de región de la estructura QTBT 130 (es decir, las líneas continuas) y elementos de sintaxis (como la información de división) para un nivel de árbol de predicción de la estructura QTBT 130 (es decir, las líneas discontinuas). El codificador de video 200 puede codificar, y el decodificador de video 300 puede decodificar datos de video, tales como datos de predicción y transformación, para CU representadas por nodos hoja terminales de la estructura QTBT 130.

[0127] En general, la CTU 132 de la FIG. 2B puede estar asociada con parámetros que definen tamaños de bloques correspondientes a nodos de la estructura QTBT 130 en el primer y segundo nivel. Estos parámetros pueden incluir un tamaño de CTU (que representa un tamaño de CTU 132 en muestras), un tamaño mínimo de *quadtree*

(MinQTSIZE, que representa un tamaño mínimo permitido de nodo hoja de *quadtree*), un tamaño máximo de árbol binario (MaxBTSIZE, que representa un tamaño máximo permitido de nodo raíz de árbol binario), una profundidad máxima del árbol binario (MaxBTDepth, que representa una profundidad máxima permitida del árbol binario) y un tamaño mínimo del árbol binario (MinBTSIZE, que representa el tamaño mínimo permitido del nodo hoja del árbol binario).

[0128] El nodo raíz de una estructura QTBT correspondiente a una CTU puede tener cuatro nodos secundarios en el primer nivel de la estructura QTBT, cada uno de los cuales puede dividirse de acuerdo con la partición de *quadtree*. Es decir, los nodos del primer nivel son nodos hoja (que no tienen nodos secundarios) o tienen cuatro nodos secundarios. El ejemplo de la estructura QTBT 130 representa tales nodos con la inclusión del nodo primario y los nodos secundarios con líneas continuas para las ramas. Si los nodos del primer nivel no son mayores que el tamaño máximo permitido del nodo raíz del árbol binario (MaxBTSIZE), pueden dividirse aún más por los respectivos árboles binarios. La división del árbol binario de un nodo puede repetirse hasta que los nodos resultantes de la división alcancen el tamaño mínimo permitido del nodo hoja del árbol binario (MinBTSIZE) o la profundidad máxima permitida del árbol binario (MaxBTDepth). El ejemplo de la estructura QTBT 130 representa tales nodos con líneas discontinuas para las ramas. El nodo de hoja de árbol binario se conoce como una unidad de codificación (CU), que se utiliza para la predicción (por ejemplo, predicción intraimagen o interimagen) y transformación, sin ninguna partición adicional. Como se discutió con anterioridad, las CU también pueden denominarse “bloques de video” o “bloques”.

[0129] En un ejemplo de la estructura de partición QTBT, el tamaño de la CTU se establece en 128x128 (muestras de luma y dos muestras de croma de 64x64 correspondientes), el MinQTSIZE se establece en 16x16, el MaxBTSIZE se establece en 64x64, el MinBTSIZE (para ancho y altura) se establece en 4, y la MaxBTDepth se establece en 4. La partición de *quadtree* se aplica primero a la CTU para generar nodos hoja de *quadtree*. Los nodos hoja de *quadtree* pueden tener un tamaño de 16x16 (es decir, el tamaño MinQTSIZE) a 128x128 (es decir, el tamaño de la CTU). Si el nodo hoja de *quadtree* es de 128x128, no se dividirá más por el árbol binario, ya que el tamaño excede el MaxBTSIZE (es decir, 64x64, en este ejemplo). De lo contrario, el árbol binario dividirá más el nodo hoja de *quadtree*. Por lo tanto, el nodo hoja de *quadtree* es también el nodo raíz del árbol binario y tiene la profundidad del árbol binario como 0. Cuando la profundidad del árbol binario alcanza MaxBTDepth (4, en este ejemplo), no se permite más división. Cuando el nodo del árbol binario tiene un ancho igual a MinBTSIZE (4, en

este ejemplo), implica que no se permite más división horizontal. De manera similar, un nodo de árbol binario que tiene una altura igual a MinBTSIZE implica que no se permite más división vertical para ese nodo de árbol binario. Como se señaló con anterioridad, los nodos hoja del árbol binario se denominan CU y se procesan más de acuerdo con la predicción y se transforman sin más particiones.

[0130] La FIG. 3 es un diagrama de bloques que ilustra un codificador de video 200 de ejemplo que puede realizar las técnicas de esta divulgación. La FIG. 3 se proporciona con fines explicativos y no debe considerarse una limitación de las técnicas como se ejemplifica y describe ampliamente en esta divulgación. Para fines de explicación, esta divulgación describe el codificador de video 200 en el contexto de estándares de codificación de video tales como el estándar de codificación de video HEVC y el estándar de codificación de video H.266 en desarrollo. Sin embargo, las técnicas de esta divulgación no se limitan a estos estándares de codificación de video y son aplicables generalmente a la codificación y decodificación de video.

[0131] En el ejemplo de la FIG. 3, el codificador de video 200 incluye la memoria de datos de video 230, la unidad de selección de modo 202, la unidad de generación residual 204, la unidad de procesamiento de transformación 206, la unidad de cuantificación 208, la unidad de cuantificación inversa 210, la unidad de procesamiento de transformación inversa 212, la unidad de reconstrucción 214, la unidad de filtro 216, el búfer de imagen decodificada (DPB) 218 y la unidad de codificación de entropía 220. Cualquiera o todos de la memoria de datos de video 230, la unidad de selección de modo 202, la unidad de generación residual 204, la unidad de procesamiento de transformación 206, la unidad de cuantificación 208, la unidad de cuantificación inversa 210, la unidad de procesamiento de transformación inversa 212, la unidad de reconstrucción 214, la unidad de filtro 216, el DPB 218 y la unidad de codificación de entropía 220 pueden implementarse en uno o más procesadores o en circuitos de procesamiento. Además, el codificador de video 200 puede incluir procesadores o circuitos de procesamiento adicionales o alternativos para realizar estas y otras funciones.

[0132] La memoria de datos de video 230 puede almacenar datos de video para ser codificados por los componentes del codificador de video 200. El codificador de video 200 puede recibir los datos de video almacenados en la memoria de datos de video 230 de, por ejemplo, la fuente de video 104 (FIG. 1). El DPB 218 puede actuar como una memoria de imagen de referencia que almacena datos de video de referencia para su uso en la predicción de datos de video posteriores por el codificador de video 200. La

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

[0133] En esta divulgación, la referencia a la memoria de datos de video 230 no debe interpretarse como limitada a la memoria interna del codificador de video 200, a menos que se describa específicamente como tal, o la memoria externa al codificador de video 200, a menos que se describa específicamente como tal. Más bien, la referencia a la memoria de datos de video 230 debe entenderse como memoria de referencia que almacena datos de video que el codificador de video 200 recibe para codificar (por ejemplo, datos de video para un bloque actual que se codificará). La memoria 106 de la FIG. 1 también puede proporcionar almacenamiento temporal de salidas de las diversas unidades del codificador de video 200.

[0134] Las diversas unidades de la FIG. 3 se ilustran para ayudar a comprender las operaciones realizadas por el codificador de video 200. Las unidades pueden implementarse como circuitos de función fija, circuitos programables o una combinación de estos. Los circuitos de función fija se refieren a circuitos que proporcionan una funcionalidad particular y están preestablecidos en las operaciones que se pueden realizar. Los circuitos programables se refieren a circuitos que se pueden programar para realizar diversas tareas y proporcionan una funcionalidad flexible en las operaciones que se pueden realizar. Por ejemplo, los circuitos programables pueden ejecutar *software* o *firmware* que hacen que los circuitos programables funcionen de la manera definida por las instrucciones del *software* o *firmware*. Los circuitos de función fija pueden ejecutar instrucciones de *software* (por ejemplo, para recibir parámetros o parámetros de salida), pero los tipos de operaciones que realizan los circuitos de función fija son generalmente inmutables. En algunos ejemplos, una o más de las unidades pueden ser bloques de circuitos distintos (función fija o programable), y en algunos ejemplos, la unidad o las unidades pueden ser circuitos integrados.

[0135] El codificador de video 200 puede incluir unidades lógicas aritméticas (ALU), unidades de funciones elementales (EFU), circuitos digitales, circuitos analógicos y/o núcleos programables, formados a partir de circuitos programables. En los ejemplos en

los que las operaciones del codificador de video 200 se realizan con el uso de un *software* ejecutado por los circuitos programables, la memoria 106 (FIG. 1) puede almacenar el código objeto del *software* que el codificador de video 200 recibe y ejecuta, u otra memoria dentro del codificador de video 200 (no mostrada) puede almacenar tales instrucciones.

[0136] La memoria de datos de video 230 está configurada para almacenar datos de video recibidos. El codificador de video 200 puede recuperar una imagen de los datos de video de la memoria de datos de video 230 y proporcionar los datos de video a la unidad de generación residual 204 y la unidad de selección de modo 202. Los datos de video en la memoria de datos de video 230 pueden ser datos de video sin procesar que se codificarán.

[0137] La unidad de selección de modo 202 incluye una unidad de estimación de movimiento 222, una unidad de compensación de movimiento 224 y una unidad de intrapredicción 226. La unidad de selección de modo 202 puede incluir unidades funcionales adicionales para realizar predicciones de video de acuerdo con otros modos de predicción. Como ejemplos, la unidad de selección de modo 202 puede incluir una unidad paleta, una unidad de copia intrabloque (que puede ser parte de la unidad de estimación de movimiento 222 y/o la unidad de compensación de movimiento 224), una unidad afín, una unidad de modelo lineal (LM), o similar.

[0138] La unidad de selección de modo 202 generalmente coordina múltiples pases de codificación para probar combinaciones de parámetros de codificación y los valores de distorsión de velocidad resultantes para tales combinaciones. Los parámetros de codificación pueden incluir la división de CTU en CU, modos de predicción para las CU, tipos de transformación para datos residuales de las CU, parámetros de cuantificación para datos residuales de las CU, etc. La unidad de selección de modo 202 puede finalmente seleccionar la combinación de parámetros de codificación que tienen valores de distorsión de velocidad que son mejores que las otras combinaciones probadas.

[0139] El codificador de video 200 puede dividir una imagen recuperada de la memoria de datos de video 230 en una serie de CTU, y encapsular una o más CTU dentro de un segmento. La unidad de selección de modo 210 puede dividir una CTU de la imagen de acuerdo con una estructura de árbol, tal como la estructura QTBT o la estructura *quadtree* de HEVC descrita con anterioridad. Como se describió con anterioridad, el codificador de video 200 puede formar una o más CU a partir de la partición de una CTU de acuerdo con la estructura de árbol. Tal CU también se puede denominar generalmente como un “bloque de video” o “bloque”.

[0140] En general, la unidad de selección de modo 202 también controla sus componentes (por ejemplo, la unidad de estimación de movimiento 222, la unidad de compensación de movimiento 224 y la unidad de intrapredicción 226) para generar un bloque de predicción para un bloque actual (por ejemplo, una CU actual, o en HEVC, la parte superpuesta de una PU y una TU). Para la interpredicción de un bloque actual, la unidad de estimación de movimiento 222 puede realizar una búsqueda de movimiento para identificar uno o más bloques de referencia que coincidan estrechamente en una o más imágenes de referencia (por ejemplo, una o más imágenes codificadas previamente almacenadas en el DPB 218). En particular, la unidad de estimación de movimiento 222 puede calcular un valor representativo de cuán similar es un bloque de referencia potencial al bloque actual, por ejemplo, de acuerdo con la suma de la diferencia absoluta (SAD), la suma de las diferencias cuadradas (SSD), la diferencia absoluta media (MAD), diferencias cuadráticas medias (MSD) o similares. La unidad de estimación de movimiento 222 generalmente puede realizar estos cálculos con el uso de diferencias muestra por muestra entre el bloque actual y el bloque de referencia que se está considerando. La unidad de estimación de movimiento 222 puede identificar un bloque de referencia que tiene un valor más bajo resultante de estos cálculos, lo que indica un bloque de referencia que coincide más estrechamente con el bloque actual.

[0141] La unidad de estimación de movimiento 222 puede formar uno o más vectores de movimiento (MV) que definen las posiciones de los bloques de referencia en las imágenes de referencia con respecto a la posición del bloque actual en una imagen actual. La unidad de estimación de movimiento 222 puede proporcionar entonces los vectores de movimiento a la unidad de compensación de movimiento 224. Por ejemplo, para la predicción unidireccional, la unidad de estimación de movimiento 222 puede proporcionar un solo vector de movimiento, mientras que, para la interpredicción bidireccional, la unidad de estimación de movimiento 222 puede proporcionar dos vectores de movimiento. La unidad de compensación de movimiento 224 puede generar entonces un bloque de predicción con el uso de los vectores de movimiento. Por ejemplo, la unidad de compensación de movimiento 224 puede recuperar datos del bloque de referencia con el uso del vector de movimiento. Como otro ejemplo, si el vector de movimiento tiene una precisión de muestra fraccional, la unidad de compensación de movimiento 224 puede interpolar valores para el bloque de predicción de acuerdo con uno o más filtros de interpolación. Además, para la interpredicción bidireccional, la unidad de compensación de movimiento 224 puede recuperar datos para dos bloques de referencia identificados por los respectivos vectores de movimiento y combinar los

datos recuperados, por ejemplo, mediante el promedio de muestra por muestra o el promedio ponderado.

[0142] Como otro ejemplo, para la intrapredicción o la codificación de intrapredicción, la unidad de intrapredicción 226 puede generar el bloque de predicción a partir de muestras vecinas al bloque actual. Por ejemplo, para modos direccionales, la unidad de intrapredicción 226 generalmente puede combinar matemáticamente valores de muestras vecinas y llenar estos valores calculados en la dirección definida a través del bloque actual para producir el bloque de predicción. Como otro ejemplo, para el modo DC, la unidad de intrapredicción 226 puede calcular un promedio de las muestras vecinas al bloque actual y generar el bloque de predicción para incluir este promedio resultante para cada muestra del bloque de predicción.

[0143] La unidad de selección de modo 202 proporciona el bloque de predicción a la unidad de generación residual 204. La unidad de generación residual 204 recibe una versión sin procesar y sin codificar del bloque actual desde la memoria de datos de video 230 y el bloque de predicción desde la unidad de selección de modo 202. Unidad de generación residual 204 calcula las diferencias muestra por muestra entre el bloque actual y el bloque de predicción. Las diferencias 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 con el uso de la modulación de código de pulso diferencial residual (RDPCM). En algunos ejemplos, la unidad de generación residual 204 puede formarse con el uso de uno o más circuitos sustractores que realizan sustracción binaria.

[0144] En los ejemplos en los que la unidad de selección de modo 202 divide las CU en PU, cada PU puede asociarse con una unidad de predicción de luma y las unidades de predicción de croma correspondientes. El codificador de video 200 y el decodificador de video 300 pueden admitir PU que tienen varios tamaños. Como se indicó con anterioridad, el tamaño de una CU puede referirse al tamaño del bloque de codificación de luma de la CU y el tamaño de una PU puede referirse al tamaño de una unidad de predicción de luma de la PU. Suponiendo que el tamaño de una CU particular es $2N \times 2N$, el codificador de video 200 puede admitir tamaños de PU de $2N \times 2N$ o $N \times N$ para intrapredicción, y tamaños de PU simétricos de $2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$ o similares para interpredicción. El codificador de video 200 y el decodificador de video 300 también pueden admitir particiones asimétricas para tamaños de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$ y $nR \times 2N$ para interpredicción.

[0145] En los ejemplos en los que la unidad de selección de modo no divide más una CU en PU, cada CU puede estar asociada con un bloque de codificación de luma y los bloques de codificación de croma correspondientes. Como se indicó con anterioridad, el tamaño de una CU puede referirse al tamaño del bloque de codificación de luma de la CU. El codificador de video 200 y el decodificador de video 120 pueden admitir tamaños CU de $2N \times 2N$, $2N \times N$ o $N \times 2N$.

[0146] Para otras técnicas de codificación de video tales como una codificación en modo de copia intrabloque, una codificación en modo afín y una codificación en modo de modelo lineal (LM), como algunos ejemplos, la unidad de selección de modo 202, a través de las unidades respectivas asociadas con las técnicas de codificación, genera un bloque de predicción para el bloque actual que se codifica. En algunos ejemplos, como la codificación de modo de paleta, la unidad de selección de modo 202 puede no generar un bloque de predicción, y en su lugar generar elementos de sintaxis que indican la manera de reconstruir el bloque en base a una paleta seleccionada. En tales modos, la unidad de selección de modo 202 puede proporcionar estos elementos de sintaxis a la unidad de codificación de entropía 220 a codificar.

[0147] Como se describió con anterioridad, la unidad de generación residual 204 recibe los datos de video para el bloque actual y el bloque de predicción correspondiente. La unidad de generación residual 204 genera entonces un bloque residual para el bloque actual. Para generar el bloque residual, la unidad de generación residual 204 calcula las diferencias muestra por muestra entre el bloque de predicción y el bloque actual.

[0148] La unidad de procesamiento de transformación 206 aplica una o más transformaciones al bloque residual para generar un bloque de coeficientes de transformación (denominado en este documento “bloque de coeficiente de transformación”). La unidad de procesamiento de transformación 206 puede aplicar diversas transformaciones a un bloque residual para formar el bloque de coeficiente de transformación. Por ejemplo, la unidad de procesamiento de transformación 206 puede aplicar una transformada de coseno discreta (DCT), una transformada direccional, una transformada de Karhunen-Loeve (KLT) o una transformada conceptualmente similar a un bloque residual. En algunos ejemplos, la unidad de procesamiento de transformación 206 puede realizar múltiples transformaciones a un bloque residual, por ejemplo, una transformada primaria y una transformada secundaria, tal como una transformada rotacional. En algunos ejemplos, la unidad de procesamiento de transformación 206 no aplica transformaciones a un bloque residual. Como se analiza en el presente documento, la unidad de procesamiento de transformación 206 puede aplicar

selectivamente diferentes transformaciones a diferentes bloques de coeficientes (es decir, bloques de coeficientes de transformación).

[0149] La unidad de cuantificación 208 puede cuantificar los coeficientes de transformación en un bloque de coeficiente de transformación, para producir un bloque de coeficiente de transformación cuantificado. La unidad de cuantificación 208 puede cuantificar los coeficientes de transformación de un bloque de coeficiente de transformación de acuerdo con un valor de parámetro de cuantificación (QP) asociado con el bloque actual. El codificador de video 200 (por ejemplo, a través de la unidad de selección de modo 202) puede ajustar el grado de cuantificación aplicado a los bloques de coeficientes asociados con el bloque actual al ajustar el valor de QP asociado con la CU. La cuantificación puede introducir pérdida de información y, por lo tanto, los coeficientes de transformación cuantificados pueden tener una precisión menor que los coeficientes de transformación originales producidos por la unidad de procesamiento de transformación 206.

[0150] La unidad de cuantificación inversa 210 y la unidad de procesamiento de transformación inversa 212 pueden aplicar cuantificación inversa y transformadas inversas a un bloque de coeficiente de transformación cuantificado, respectivamente, para reconstruir un bloque residual del bloque de coeficiente de transformación. La unidad de reconstrucción 214 puede producir un bloque reconstruido correspondiente al bloque actual (aunque potencialmente con cierto grado de distorsión) basado en el bloque residual reconstruido y un bloque de predicción generado por la unidad de selección de modo 202. Por ejemplo, la unidad de reconstrucción 214 puede agregar muestras del bloque residual reconstruido a muestras correspondientes del bloque de predicción generado por la unidad de selección de modo 202 para producir el bloque reconstruido.

[0151] La unidad de filtro 216 puede realizar una o más operaciones de filtro en bloques reconstruidos. Por ejemplo, la unidad de filtro 216 puede realizar operaciones de desbloqueo para reducir los artefactos de bloqueo a lo largo de los bordes de las CU. Las operaciones de la unidad de filtro 216 pueden omitirse, en algunos ejemplos.

[0152] El codificador de video 200 almacena bloques reconstruidos en el DPB 218. Por ejemplo, en ejemplos donde no se necesitan operaciones de la unidad de filtro 224, la unidad de reconstrucción 214 puede almacenar bloques reconstruidos en el DPB 218. En ejemplos donde se necesitan operaciones de la unidad de filtro 224, la unidad de filtro 216 puede almacenar los bloques reconstruidos filtrados en el DPB 218. La unidad de estimación de movimiento 222 y la unidad de compensación de movimiento 224

pueden recuperar una imagen de referencia del DPB 218, formada a partir de los bloques reconstruidos (y potencialmente filtrados), para interprededir los bloques de imágenes codificadas posteriormente. Además, la unidad de intrapredicción 226 puede usar bloques reconstruidos en el DPB 218 de una imagen actual para intraprededir otros bloques en la imagen actual.

[0153] En general, la unidad de codificación de 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 de entropía 220 puede codificar por entropía bloques de coeficientes de transformación cuantificados de la unidad de cuantificación 208. Como otro ejemplo, la unidad de codificación de entropía 220 puede codificar por entropía elementos de sintaxis de predicción (por ejemplo, información de movimiento para interpredicción o información intramodal para intrapredicción) desde la unidad de selección de modo 202. La unidad de codificación de entropía 220 puede realizar una o más operaciones de codificación de entropía en los elementos de sintaxis, que son otro ejemplo de datos de video, para generar datos codificados por entropía. Por ejemplo, la unidad de codificación de entropía 220 puede realizar una operación de codificación de longitud variable adaptativa al contexto (CAVLC), una operación de CABAC, una operación de codificación de longitud variable a variable (V2V), una codificación aritmética binaria adaptativa al contexto basada en sintaxis (SBAC), una operación de codificación de Entropía de Partición de Intervalo de Probabilidad (PIPE), una operación de codificación exponencial-Golomb u otro tipo de operación de codificación de entropía en los datos. En algunos ejemplos, la unidad de codificación de entropía 220 puede funcionar en modo de derivación donde los elementos de sintaxis no están codificados por entropía.

[0154] El codificador de video 200 puede emitir un flujo de bits que incluye los elementos de sintaxis codificados por entropía necesarios para reconstruir bloques de un segmento o imagen. En particular, la unidad de codificación de entropía 220 puede emitir el flujo de bits.

[0155] Las operaciones descritas con anterioridad se describen con respecto a un bloque. Dicha descripción debe entenderse como operaciones para un bloque de codificación de luma y/o bloques de codificación de croma. Como se describió con anterioridad, en algunos ejemplos, el bloque de codificación de luma y los bloques de codificación de croma son componentes de luma y croma de una CU. En algunos ejemplos, el bloque de codificación de luma y los bloques de codificación de croma son componentes de luma y croma de una PU.

[0156] En algunos ejemplos, las operaciones realizadas con respecto a un bloque de codificación de luma no necesitan repetirse para los bloques de codificación de croma. Como un ejemplo, las operaciones para identificar un vector de movimiento (MV) y una imagen de referencia para un bloque de codificación de luma no necesitan repetirse para identificar un MV y una imagen de referencia para los bloques de croma. Por el contrario, el MV para el bloque de codificación de luma se puede escalar para determinar el MV para los bloques de croma, y la imagen de referencia puede ser la misma. Como otro ejemplo, el proceso de intrapredicción puede ser el mismo para los bloques de codificación de luma y los bloques de codificación de croma.

[0157] El codificador de video 200 representa un ejemplo de un dispositivo configurado para codificar datos de video que incluye una memoria configurada para almacenar datos de video, y una o más unidades de procesamiento implementadas en circuitos y configuradas para derivar, para un bloque de coeficiente actual de un bloque de video, un tipo de transformación de una pluralidad de tipos de transformación. El codificador de video puede transformar, con el uso del tipo de transformación seleccionado, el bloque de coeficiente actual para obtener un bloque de datos residuales reconstruidos para el bloque de video; y reconstruir, en función de los datos residuales reconstruidos para el bloque de video, el bloque de video.

[0158] La FIG. 4 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. 4 se proporciona con fines de explicación y no limita las técnicas como se ejemplifica y describe ampliamente en esta divulgación. Para fines de explicación, esta divulgación describe el decodificador de video 300 de acuerdo con las técnicas de JEM, VVC y HEVC. Sin embargo, las técnicas de la presente divulgación pueden realizarse mediante dispositivos de codificación de video que están configurados de acuerdo con otros estándares de codificación de video.

[0159] En el ejemplo de la FIG. 4, el decodificador de video 300 incluye la memoria de búfer de imagen codificada (CPB) 320, la unidad de decodificación de entropía 302, la unidad de procesamiento de predicción 304, la unidad de cuantificación inversa 306, la unidad de procesamiento de transformada inversa 308, la unidad de reconstrucción 310, la unidad de filtro 312 y el búfer de imagen decodificada (DPB) 314. Cualquiera o todos de la memoria CPB 320, la unidad de decodificación de entropía 302, la unidad de procesamiento de predicción 304, la unidad de cuantificación inversa 306, la unidad de procesamiento de transformación inversa 308, la unidad de reconstrucción 310, la unidad de filtro 312 y el DPB 314 pueden implementarse en uno o más procesadores o

en circuitos de procesamiento. Además, el decodificador de video 300 puede incluir procesadores o circuitos de procesamiento adicionales o alternativos para realizar estas y otras funciones.

[0160] La unidad de procesamiento de predicción 304 incluye la unidad de compensación de movimiento 316 y la unidad de intrapredicción 318. La unidad de procesamiento de predicción 304 puede incluir unidades de adición para realizar la predicción de acuerdo con otros modos de predicción. Como ejemplos, la unidad de procesamiento de predicción 304 puede incluir una unidad de paleta, una unidad de copia intrabloque (que puede formar parte de la unidad de compensación de movimiento 318), una unidad afín, una unidad de modelo lineal (LM) o similares. En otros ejemplos, el decodificador de video 300 puede incluir más, menos o diferentes componentes funcionales.

[0161] La memoria CPB 320 puede almacenar datos de video, tales como un flujo de bits de video codificado, para ser decodificados por los componentes del decodificador de video 300. Los datos de video almacenados en la memoria CPB 320 pueden obtenerse, por ejemplo, del medio legible por computadora 110 (FIG. 1). La memoria CPB 320 puede incluir un CPB que almacena datos de video codificados (por ejemplo, elementos de sintaxis) de un flujo de bits de video codificado. Además, la memoria CPB 320 puede almacenar datos de video que no sean elementos de sintaxis de una imagen codificada, como datos temporales que representan salidas de las diversas unidades del decodificador de video 300. El DPB 314 generalmente almacena imágenes decodificadas, que el decodificador de video 300 puede generar y/o usar como datos de video de referencia al decodificar datos o imágenes posteriores del flujo de bits de video codificado. La memoria CPB 320 y el DPB 314 pueden estar formadas por cualquiera de una variedad de dispositivos de memoria, como la memoria dinámica de acceso aleatorio (DRAM), incluida la DRAM síncrona (SDRAM), la RAM magnetorresistiva (MRAM), la RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. La memoria CPB 320 y el DPB 314 pueden ser proporcionadas por el mismo dispositivo de memoria o dispositivos de memoria separados. En varios ejemplos, la memoria CPB 320 puede estar en chip con otros componentes del decodificador de video 300, o fuera de chip en relación con esos componentes.

[0162] Adicional o alternativamente, en algunos ejemplos, el decodificador de video 300 puede recuperar datos de video codificados de la memoria 120 (FIG. 1). Es decir, la memoria 120 puede almacenar datos como se discutió con anterioridad con la memoria CPB 320. Del mismo modo, la memoria 120 puede almacenar instrucciones para ser

ejecutadas por el decodificador de video 300, cuando parte o la totalidad de la funcionalidad del decodificador de video 300 se implementa en el *software* para ser ejecutado por circuitos de procesamiento del decodificador de video 300.

[0163] Las diversas unidades mostradas en la FIG. 4 se ilustran para ayudar a comprender las operaciones realizadas por el decodificador de video 300. Las unidades pueden implementarse como circuitos de función fija, circuitos programables o una combinación de estos. Similar a la FIG. 3, los circuitos de función fija se refieren a circuitos que proporcionan una funcionalidad particular, y están preestablecidos en las operaciones que se pueden realizar. Los circuitos programables se refieren a circuitos que se pueden programar para realizar diversas tareas y proporcionan una funcionalidad flexible en las operaciones que se pueden realizar. Por ejemplo, los circuitos programables pueden ejecutar *software* o *firmware* que hacen que los circuitos programables funcionen de la manera definida por las instrucciones del *software* o *firmware*. Los circuitos de función fija pueden ejecutar instrucciones de *software* (por ejemplo, para recibir parámetros o parámetros de salida), pero los tipos de operaciones que realizan los circuitos de función fija son generalmente inmutables. En algunos ejemplos, una o más de las unidades pueden ser bloques de circuitos distintos (función fija o programable), y en algunos ejemplos, la unidad o las unidades pueden ser circuitos integrados.

[0164] El decodificador de video 300 puede incluir ALU, EFU, circuitos digitales, circuitos analógicos y/o núcleos programables formados a partir de circuitos programables. En los ejemplos en los que las operaciones del decodificador de video 300 se realizan mediante *software* que se ejecuta en los circuitos programables, la memoria en chip o fuera de chip puede almacenar instrucciones (por ejemplo, código de objeto) del *software* que el decodificador de video 300 recibe y ejecuta.

[0165] La unidad de decodificación de entropía 302 puede recibir datos de video codificados del CPB y decodificar por entropía los datos de video para reproducir elementos de sintaxis. La unidad de procesamiento de predicción 304, la unidad de cuantificación inversa 306, la unidad de procesamiento de transformación inversa 308, la unidad de reconstrucción 310 y la unidad de filtro 312 pueden generar datos de video decodificados basados en los elementos de sintaxis extraídos del flujo de bits.

[0166] En general, el decodificador de video 300 reconstruye una imagen bloque por bloque. El decodificador de video 300 puede realizar una operación de reconstrucción en cada bloque individualmente (donde el bloque que se está reconstruyendo actualmente, es decir, decodificado, puede denominarse un “bloque actual”).

[0167] La unidad de decodificación de entropía 302 puede decodificar por entropía elementos de sintaxis que definen coeficientes de transformación cuantificados de un bloque de coeficiente de transformación cuantificado, así como información de transformación, tal como un parámetro de cuantificación (QP) y/o indicaciones de modo de transformación. La unidad de cuantificación inversa 306 puede usar el QP asociado con el bloque de coeficiente de transformación cuantificado para determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa para aplicar la unidad de cuantificación inversa 306. La unidad de cuantificación inversa 306 puede, por ejemplo, realizar una operación de desplazamiento a la izquierda bit a bit para cuantificar inversamente los coeficientes de transformación cuantificados. La unidad de cuantificación inversa 306 puede formar así un bloque de coeficiente de transformación que incluye coeficientes de transformación.

[0168] Después de que la unidad de cuantificación inversa 306 forma el bloque de coeficiente de transformación, la unidad de procesamiento de transformación inversa 308 puede aplicar una o más transformaciones inversas al bloque de coeficiente de transformación para generar un bloque residual asociado con el bloque actual. Por ejemplo, la unidad de procesamiento de transformación inversa 308 puede aplicar una DCT inversa, una transformada entera inversa, una transformada de Karhunen–Loeve inversa (KLT), una transformada rotacional inversa, una transformada direccional inversa u otra transformación inversa al bloque de coeficientes. Como se analiza en el presente documento, la unidad de procesamiento de transformación 206 puede aplicar selectivamente diferentes transformaciones a diferentes bloques de coeficientes (es decir, bloques de coeficientes de transformación).

[0169] Además, la unidad de procesamiento de predicción 304 genera un bloque de predicción de acuerdo con los elementos de sintaxis de información de predicción que fueron decodificados por entropía por la unidad de decodificación de entropía 302. Por ejemplo, si los elementos de sintaxis de información de predicción indican que el bloque actual es inter-predicho, 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 el DPB 314 desde la cual recuperar un bloque de referencia, así como un vector de movimiento que identifica una ubicación del bloque de referencia en la imagen de referencia en relación con la ubicación del bloque actual en la imagen actual. La unidad de compensación de movimiento 316 generalmente puede realizar el proceso de interpredicción de una manera que es sustancialmente similar a la descrita con respecto a la unidad de

compensación de movimiento 224 (FIG. 3).

[0170] Como otro ejemplo, si los elementos de sintaxis de información de predicción indican que el bloque actual es intrapredicho, la unidad de interpredicción 318 puede generar el bloque de predicción de acuerdo con un modo de intrapredicción indicado por los elementos de sintaxis de información de predicción. Nuevamente, la unidad de intrapredicción 318 generalmente puede realizar el proceso de intrapredicción de una manera que es sustancialmente similar a la descrita con respecto a la unidad de intrapredicción 226 (FIG. 3). La unidad de interpredicción 318 puede recuperar datos de muestras vecinas al bloque actual del DPB 314.

[0171] La unidad de reconstrucción 310 puede reconstruir el bloque actual con el uso del bloque de predicción y el bloque residual. Por ejemplo, la unidad de reconstrucción 310 puede agregar muestras del bloque residual a las muestras correspondientes del bloque de predicción para reconstruir el bloque actual.

[0172] La unidad de filtro 312 puede realizar una o más operaciones de filtro en bloques reconstruidos. Por ejemplo, la unidad de filtro 312 puede realizar operaciones de desbloqueo para reducir los artefactos de bloqueo a lo largo de los bordes de los bloques reconstruidos. Las operaciones de la unidad de filtro 312 no se realizan necesariamente en todos los ejemplos.

[0173] El decodificador de video 300 puede almacenar los bloques reconstruidos en el DPB 314. Como se discutió con anterioridad, el 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 posterior, a la unidad de procesamiento de predicción 304. Además, el decodificador de video 300 puede emitir imágenes decodificadas desde el DPB para su posterior presentación en un dispositivo de visualización, tal como el dispositivo de visualización 118 de la FIG. 1.

[0174] 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 derivar, para un bloque de coeficiente actual de un bloque de video, un tipo de transformación de una pluralidad de tipos de transformación. El codificador de video puede transformar, con el uso del tipo de transformación seleccionado, el bloque de coeficiente actual para obtener un bloque de datos residuales reconstruidos para el bloque de video; y reconstruir, en función de los datos residuales reconstruidos para el bloque de video, el bloque de video.

[0175] La FIG. 5 es un diagrama de bloques que ilustra un sistema para codificación de

video híbrido con selección de transformación adaptativa. Puede considerarse que el codificador de video 200' de la FIG. 5 ilustra un sistema de codificación de video similar al codificador 200 de video de las FIG. 1 y 3. Por ejemplo, puede considerarse que la predicción de bloque 202', la transformación de bloque 206', la cuantificación 208', la cuantificación inversa 210', la transformación inversa 212', el búfer de cuadro 218' y la codificación de entropía 220' del codificador de video 200' realizan operaciones similares a la unidad de selección de modo 202, la unidad de procesamiento de transformación 206, la unidad de cuantificación 208, la unidad de cuantificación inversa 210, la unidad de procesamiento de transformación inversa 212, el búfer de imagen decodificada 218 y la unidad de codificación de entropía 220 del codificador de video 200 de la FIG. 3. Como se muestra en la FIG. 5, el codificador de video 200' puede incluir el banco de transformación 207', que puede configurarse para funcionar junto con la transformación de bloque 206' para transformar los datos residuales. Por ejemplo, el banco de transformación 207' y la transformación de bloque 206' pueden seleccionar y realizar colectivamente varias transformaciones (por ejemplo, varias DCT o DST) para cada bloque de residuos de predicción. Como se discutió con anterioridad, en algunos ejemplos, el banco de transformación 207' y la transformación de bloque 206' pueden indicar la elección de transformación como información secundaria. Por ejemplo, la transformación de bloque 206' puede hacer que la codificación de entropía 220' codifique un elemento de sintaxis que indique explícitamente la transformación utilizada (es decir, t).

[0176] En algunos ejemplos, el banco de transformación 207' y la transformación de bloque 206' pueden calcular las transformaciones de bloque de una manera separable. Por ejemplo, para reducir la complejidad del cálculo, el banco de transformación 207' y la transformación de bloque 206' pueden transformar las líneas horizontales y verticales de forma independiente como se muestra en la FIG. 6. En otras palabras, las muestras a lo largo de las flechas horizontal y vertical de la FIG. 6 pueden transformarse de forma independiente.

[0177] En los estándares de codificación de video anteriores a HEVC, solo se usa una transformada separable fija donde se usa DCT-2 tanto vertical como horizontalmente. En HEVC, además de DCT-2, DST-7 también se emplea para bloques 4x4 como una transformada separable fija. Los documentos US-2016-0219290-A1 y US-2018-0020218-A1 describen extensiones adaptativas de esas transformadas fijas, y se adoptó un ejemplo de AMT en US-2016-0219290-A1 en el *Joint Experimental Model (JEM)* del *Joint Video Experts Team (JVET)*, *Joint Video Experts Team (JVET)* de ITU-T SG 16 WP

3 e ISO/IEC JTC 1/SC 29/WG 11, JEM Software, https://jvet.hhi.fraunhofer.de/svn/svn_HMJEMSoftware/tags/HM-16.6-JEM-7.0.

[0178] De acuerdo con una o más técnicas de esta divulgación, un codificador de video (por ejemplo, un codificador de video y/o un decodificador de video) puede realizar una selección de transformación implícita. Por ejemplo, el codificador de video puede aplicar uno o más conjuntos de reglas para seleccionar implícitamente una transformación para transformar datos residuales para un bloque. De esta manera, el codificador de video puede mejorar la eficiencia de la codificación. En particular, las técnicas de esta divulgación permiten que el decodificador de video obtenga los beneficios de usar la selección de transformación adaptativa sin la sobrecarga de señalar realmente la transformación seleccionada.

[0179] En el Borrador 4 de VVC, hay dos derivaciones de transformación implícitas que son relativamente complicadas y no proporcionan un buen rendimiento de codificación. Esta divulgación propuso derivaciones alternativas más simples que pueden proporcionar un rendimiento/eficiencia de compresión/codificación similar o incluso mejor.

[0180] Las técnicas relacionadas en el Borrador 4 de VVC y el *software* de referencia VTM-4.0) se describen a continuación.

[0181] En el Borrador 4 de VVC/VTM-4.0, la selección de transformación múltiple (MTS) usa un indicador de alto nivel para determinar si la transformación (i) se señala explícitamente para seleccionar entre múltiples candidatos, o (ii) se deriva implícitamente en base a la forma del bloque. En el último caso, las combinaciones de DST-7 y DCT-2 como transformadas horizontales o verticales hasta el tamaño 16. Específicamente, las siguientes condiciones dependientes de la forma del bloque definen el MTS implícito en VTM-4.0:

- Si el ancho y la altura de un bloque son iguales y ambos son menores o iguales a 16, se utiliza DST-7 tanto en dirección horizontal como vertical.
- Si el ancho de un bloque es menor que su altura y es menor o igual a 16, DST-7 está en horizontal y DCT-2 se usa en dirección vertical.
- Si la altura de un bloque es menor que su ancho y es menor o igual a 16, DST-7 está en vertical y DCT-2 se usa en dirección horizontal.
- De lo contrario, DCT-2 se utiliza en ambas direcciones.

[0182] En el Borrador 4 de VVC/VTM-4.0, cuando se usa la partición intrasub-bloque (ISP) para codificar bloques de luma, se realiza una selección de transformación dependiente del modo, donde las transformadas horizontales y verticales (trTypeHor y

trTypeVer) se derivan en base a la siguiente tabla en el Borrador 4 de VVC.

Tabla: especificación de trTypeHor y trTypeVer según predModelIntra

predModelIntra	trTypeHor	trTypeVer
INTRA_PLANAR, INTRA_ANGULAR31, INTRA_ANGULAR32, INTRA_ANGULAR34, INTRA_ANGULAR36, INTRA_ANGULAR37	(nTbW >= 4 && nTbW <= 16) ? 1 : 0	(nTbH >= 4 && nTbH <= 16) ? 1 : 0
INTRA_ANGULAR33, INTRA_ANGULAR35	0	0
INTRA_ANGULAR2, INTRA_ANGULAR4,...,INTRA_ANGULAR28, INTRA_ANGULAR30, INTRA_ANGULAR39, INTRA_ANGULAR41,...,INTRA_ANGULAR63, INTRA_ANGULAR65	(nTbW >= 4 && nTbW <= 16) ? 1 : 0	0
INTRA_ANGULAR3, INTRA_ANGULAR5,..., INTRA_ANGULAR27, INTRA_ANGULAR29, INTRA_ANGULAR38, INTRA_ANGULAR40,...,INTRA_ANGULAR64, INTRA_ANGULAR66	0	(nTbH >= 4 && nTbH <= 16) ? 1 : 0

[0183] Como se discutió con anterioridad y de acuerdo con una o más técnicas de esta divulgación, un codificador de video puede aplicar uno o más conjuntos de reglas para derivar implícitamente la selección de transformación en base a la información secundaria disponible.

[0184] Como primer ejemplo, un codificador de video puede determinar que una unidad de codificación/unidad de transformación (CU/TU) se codifica con el uso de DST-7 solo en ciertas condiciones. Por ejemplo, si el tamaño máximo de transformación 1-D permitido es N en un códec, el codificador de video puede determinar que DST-7 puede usarse para todos los tamaños posibles. Por ejemplo, para un bloque NxM dado (como se muestra en la FIG. 7 cuyas N filas tienen M muestras cada una, y sus M columnas tienen N muestras), el codificador de video puede determinar que la DST-7 de N puntos puede usarse verticalmente y la DST-7 de M puntos se puede utilizar horizontalmente.

[0185] Como segundo ejemplo, para un conjunto seleccionado de dimensiones, el codificador de video puede determinar que se pueden usar diferentes combinaciones de DST-7 y DCT-2. Por ejemplo, el codificador de video puede determinar que DST-7 se

puede aplicar para cualquier fila o columna con menos o igual a K muestras, mientras que DCT-2 se puede usar para transformar cualquier fila o columna con un número de muestras mayor que K. Por ejemplo, en el ejemplo de la FIG. 7, si N es menor que K y M es mayor que K, el codificador de video puede determinar usar DST-7 de N puntos verticalmente y DCT-2 de M puntos horizontalmente. También en el ejemplo de la FIG. 7, si tanto N como M son más pequeños que K, el codificador de video puede determinar usar DST-7 tanto horizontal como verticalmente.

[0186] Como tercer ejemplo, si una CU/TU está particionada, el decodificador de video puede determinar que todas las particiones pueden usar el mismo esquema de selección de transformación implícita. En algunos ejemplos, el codificador de video puede utilizar DST-7 para todos los sub-bloques particionados (sub TU o sub CU). En algunos ejemplos, el codificador de video puede usar combinaciones de DST-7 y DCT-2 dependiendo de las dimensiones del bloque después de la partición. En algunos ejemplos, para codificar bloques que utilizan la partición intrasub-bloque (ISP) en VVC (VTM-4.0), el codificador de video puede utilizar combinaciones de DST-7 y DCT-2 dependiendo de las dimensiones del bloque como se discutió con anterioridad en el segundo ejemplo. Por ejemplo, para cualquier fila o columna con menos de 16 muestras o igual a 16 muestras, el codificador de video puede DST-7. De lo contrario, el codificador de video puede usar DCT-2 para transformar cualquier fila o columna con un número de muestras superior a 16. En algunos ejemplos, como el ISP puede tener filas/columnas con dos muestras, el codificador de video puede usar DST-7 de 2 puntos. En estándares anteriores, no se ha utilizado DST-7 de 2 puntos. Como tal, el codificador de video puede usar entradas modificadas de la matriz DST-7 de 2 puntos de la siguiente manera:

$$\{ 48, 77 \}$$

$$\{ 77, -48 \}$$

[0187] Como cuarto ejemplo, el codificador de video puede derivar la transformada basándose en modos de intrapredicción (los modos se ilustran en la Figura 8). Para los modos intraplanares y DC, el codificador de video puede usar DST-7 tanto en dirección horizontal como vertical. Para el modo angular intradiagonal (índice de modo 34 en la figura 8), el codificador de video puede utilizar DST-7 tanto en dirección horizontal como vertical. Para los modos angulares, indexados de 2 a 66, el codificador de video puede aplicar diferentes combinaciones de DST/DCT a cierto rango de modos, como intervalos predefinidos de índices de modo entre índices de modo [2, 3,..., 65, 66].

- 1) El rango de intervalos que consta de todos los modos angulares [2,3,..., 66] se

puede definir de la siguiente manera para un entero T dado entre 2 y 30:

- a. $R_1 = [2, \dots, (33 - T)]$
- b. $R_2 = [(34 - T), \dots, (34 + T)]$
- c. $R_3 = [(35 + T), \dots, 66]$

- 2) DST-7 se puede aplicar tanto horizontal como verticalmente para los modos angulares en el rango R_2 .
- 3) DST-7 se puede aplicar horizontalmente y DCT-2 verticalmente para los modos angulares en el rango R_1 .
- 4) DCT-2 se puede aplicar horizontalmente y DST-7 verticalmente para los modos angulares en el rango R_3 .

[0188] Como quinto ejemplo, además de DST-7 y DCT-2, el codificador de video puede aplicar combinaciones de diferentes tipos de DCT/DST (por ejemplo, DST-4 y DCT-8) y transformada de identidad 1-D.

[0189] Como sexto ejemplo, el codificador de video puede aplicar una o más combinaciones de los ejemplos anteriores solo para CU/TU intrapredichas.

[0190] Como séptimo ejemplo, el codificador de video puede aplicar una o más combinaciones de los ejemplos anteriores solo para CU/TU interpredichas.

[0191] Como octavo ejemplo, el codificador de video puede aplicar una o más combinaciones de los ejemplos anteriores usados para CU/TU tanto intrapredichas como interpredichas.

[0192] Como noveno ejemplo, el codificador de video puede aplicar una o más combinaciones de los ejemplos anteriores utilizados para los canales de luma o croma o ambos canales de luma y croma.

[0193] La FIG. 9 es un diagrama de flujo que ilustra un método de ejemplo para codificar un bloque actual. El bloque actual puede comprender una CU actual. Aunque se describe con respecto al codificador de video 200 (Figuras 1 y 3), debe entenderse que se pueden configurar otros dispositivos para realizar un método similar al de la FIG. 9. Por ejemplo, el codificador de video 200' de la FIG. 5 puede realizar un método similar al de la FIG. 9.

[0194] En este ejemplo, el codificador de video 200 predice inicialmente el bloque actual (350). Por ejemplo, el codificador de video 200 puede formar un bloque de predicción para el bloque actual. El codificador de video 200 puede entonces calcular un bloque residual para el bloque actual (352). Para calcular el bloque residual, el codificador de video 200 puede calcular una diferencia entre el bloque no codificado original y el bloque de predicción para el bloque actual. El codificador de video 200 puede entonces

transformar y cuantificar los coeficientes del bloque residual (354). Como se discutió con anterioridad, el codificador de video 200 puede derivar implícitamente un tipo de transformación para usar al transformar los coeficientes del bloque residual. Por ejemplo, el codificador de video 200 puede derivar el tipo de transformación con el uso de la técnica que se describe a continuación con referencia a la FIG. 11.

[0195] A continuación, el codificador de video 200 puede escanear los coeficientes de transformación cuantificados del bloque residual (356). Durante la exploración, o después de la exploración, el codificador de video 200 puede codificar por entropía los coeficientes (358). Por ejemplo, el codificador de video 200 puede codificar los coeficientes mediante CAVLC o CABAC. El codificador de video 200 puede generar entonces los datos codificados por entropía del bloque (360).

[0196] La FIG. 10 es un diagrama de flujo que ilustra un método de ejemplo para decodificar un bloque actual de datos de video. El bloque actual puede comprender una CU actual. Aunque se describe con respecto al decodificador de video 300 (FIG. 1 y 4), debe entenderse que se pueden configurar otros dispositivos para realizar un método similar al de la FIG. 10.

[0197] El decodificador de video 300 puede recibir datos codificados por entropía para el bloque actual, tales como información de predicción codificada por entropía y datos codificados por entropía para coeficientes de un bloque residual correspondiente al bloque actual (370). El decodificador de video 300 puede decodificar por entropía los datos codificados por entropía para determinar la información de predicción para el bloque actual y reproducir los coeficientes del bloque residual (372). El decodificador de video 300 puede predecir el bloque actual (374), por ejemplo, con el uso de un modo de intrapredicción o interpredicción como lo indica la información de predicción para el bloque actual, para calcular un bloque de predicción para el bloque actual. El decodificador de video 300 puede entonces escanear inversamente los coeficientes reproducidos (376), para crear un bloque de coeficientes de transformación cuantificados. El decodificador de video 300 puede entonces cuantificar y transformar inversamente los coeficientes para producir un bloque residual (378). Como se discutió con anterioridad, el decodificador de video 300 puede derivar implícitamente un tipo de transformación para usar al transformar los coeficientes del bloque residual. Por ejemplo, el decodificador de video 300 puede derivar el tipo de transformación con el uso de la técnica discutida a continuación con referencia a la FIG. 11. El decodificador de video 300 puede finalmente decodificar el bloque actual al combinar el bloque de predicción y el bloque residual (380).

[0198] La FIG. 11 es un diagrama de flujo que ilustra un método de ejemplo para inferir un tipo de transformación para un bloque de transformación de un bloque de video, de acuerdo con una o más técnicas de esta divulgación. Las técnicas de la FIG. 11 pueden realizarse mediante un codificador de video (por ejemplo, un codificador de video 200 y/o un decodificador de video 300).

[0199] Un codificador de video puede obtener un bloque de transformación actual de un bloque de video actual (1102). El bloque de transformación puede ser una matriz de coeficientes de transformación que se construye en base a uno o más elementos de sintaxis decodificados de un flujo de bits de video (por ejemplo, los elementos de sintaxis incluidos en la tabla de sintaxis de codificación residual del Borrador 4 de VVC. El bloque de video actual puede ser una unidad de codificación (CU).

[0200] El codificador de video puede inferir un tipo de transformación a partir de una pluralidad de tipos de transformación para el bloque de transformación actual. La pluralidad de tipos de transformación puede incluir una o más transformadas de coseno discretas (DCT) y una o más transformadas de seno discretas (DST).

[0201] Como se discutió con anterioridad, el codificador de video puede inferir el tipo de transformación en función de uno o más factores, tales como si el bloque de video actual está particionado mediante ISP y/o un tamaño del bloque de transformación. Como se muestra en la FIG. 11, el codificador de video puede determinar que el bloque de video actual está particionado mediante ISP (1104). El codificador de video puede determinar que el bloque de video actual está particionado mediante ISP en base a los valores de uno o más elementos de sintaxis (por ejemplo, `sps_isp_enabled_flag`, `intra_subpartitions_mode_flag` y/o `intra_subpartitions_split_flag`). Por ejemplo, en función del elemento de sintaxis `intra_subpartitions_split_flag`, el codificador de video puede determinar si el bloque de video actual no está particionado (por ejemplo, no dividido), está particionado horizontal o verticalmente.

[0202] En respuesta a la determinación de que el bloque de video actual está particionado mediante ISP (1104), el codificador de video puede determinar un tamaño del bloque de transformación actual (1106). Por ejemplo, el codificador de video puede determinar una anchura y/o una altura del bloque de transformación. En algunos ejemplos, el codificador de video puede determinar por separado el tamaño del bloque de transformación para cada subpartición. En otros ejemplos, el codificador de video puede determinar el tamaño del bloque de transformación para una sola partición y utilizar el tamaño determinado para cada partición de la unidad de codificación.

[0203] El codificador de video puede determinar si el tamaño del bloque de

transformación actual satisface un umbral de tamaño. Por ejemplo, como se muestra en la FIG. 11, el codificador de video puede determinar si el tamaño del bloque de transformación actual es mayor que un límite inferior y menor que un límite superior (es decir, si ambos ($\text{tamaño} > \text{límite inferior}$) y ($\text{tamaño} < \text{límite superior}$) son verdaderos) (1108). Como se discutió con anterioridad, en algunos ejemplos, el límite inferior puede ser 4 muestras y el límite superior puede ser 16 muestras).

[0204] En respuesta a determinar que el tamaño del bloque de transformación actual satisface el umbral de tamaño y que el bloque de video actual está particionado mediante ISP, el codificador de video puede seleccionar una DST particular de una o más DST como el tipo de transformación seleccionado. Por ejemplo, como se muestra en la FIG. 11, en respuesta a determinar que el tamaño del bloque de transformación actual satisface el umbral de tamaño y que el bloque de video actual está particionado mediante ISP, el codificador de video puede seleccionar DST-7 como el tipo de transformación inferida para el bloque de transformación actual (“Sí” ramificación de 1108, 1110). Alternativamente, en respuesta a determinar que el tamaño del bloque de transformación actual no satisface el umbral de tamaño y que el bloque de video actual está particionado mediante ISP, el codificador de video puede seleccionar DCT-2 como el tipo de transformación inferida para el bloque de transformación actual (“No” ramificación de 1108, 1112).

[0205] El codificador de video puede transformar, con el uso del tipo de transformación seleccionado, el bloque de transformación actual para obtener un bloque de datos residuales reconstruidos para el bloque de video (1114). Por ejemplo, cuando el tipo de transformación seleccionado es DST-7, el codificador de video (por ejemplo, la unidad de procesamiento de transformación inversa 212/212' del codificador de video 200/200' y/o la unidad de procesamiento de transformación inversa 308 del decodificador de video 300) puede transformar los coeficientes del bloque de transformación en los datos residuales reconstruidos al aplicar una transformada DST-7 inversa.

[0206] El codificador de video puede reconstruir, en base a los datos residuales reconstruidos para el bloque de video, el bloque de video (1116). Por ejemplo, el codificador de video puede agregar los datos residuales a un bloque de muestras intrapredichas para el bloque actual. Cuando el bloque de video se particiona mediante ISP, el codificador de video puede agregar un bloque respectivo de datos residuales reconstruidos a un bloque respectivo de muestras intrapredichas para cada subpartición respectiva del bloque de video actual.

[0207] Los siguientes ejemplos numerados pueden ilustrar uno o más aspectos de la

divulgación:

[0208] Ejemplo 1. Un método de codificación de datos de video, que comprende: derivar, para un bloque de coeficiente actual de un bloque de video, un tipo de transformación a partir de una pluralidad de tipos de transformación; transformar, con el uso del tipo de transformación seleccionado, el bloque de coeficiente actual para obtener un bloque de datos residuales reconstruidos para el bloque de video; y reconstruir, en función de los datos residuales reconstruidos para el bloque de video, el bloque de video.

[0209] Ejemplo 2. El método del ejemplo 1, en donde la pluralidad de tipos de transformación incluye una o más transformadas de coseno discretas (DCT) y/o una o más transformadas de seno discretas (DST).

[0210] Ejemplo 3. El método del ejemplo 2, en donde una o más DCT incluyen uno o más de una DCT-1, una DCT-2, una DCT-3, una DCT-4, una DCT-5, una DCT-6, una DCT-7 y una DCT-8.

[0211] Ejemplo 4. El método de cualquiera de los ejemplos 2 y 3, en donde una o más DST incluyen una o más de una DST-1, una DST-2, una DST-3, una DST-4, una DST-5, una DST-6, una DST-7 y una DST-8.

[0212] Ejemplo 5. El método de cualquiera de los ejemplos 1 a 4, en donde derivar el tipo de transformación comprende derivar el tipo de transformación en función de un tamaño del bloque de coeficientes actual.

[0213] Ejemplo 6. El método del ejemplo 5, en donde derivar el tipo de transformación basado en el tamaño del bloque de coeficiente actual comprende seleccionar el tipo de transformada DST-7 donde un tamaño máximo de transformada 1-D permitido es N.

[0214] Ejemplo 7. El método del ejemplo 6, en donde el bloque de coeficiente actual tiene dimensiones de NxM, y en donde seleccionar el tipo de transformada DST-7 comprende seleccionar una transformada DST-7 de N puntos para uso vertical y seleccionar una transformada de DST-7 de M puntos para uso horizontal.

[0215] Ejemplo 8. El método de cualquier combinación de los ejemplos 1-7, en donde derivar el tipo de transformación comprende seleccionar diferentes combinaciones de la transformada DST-7 y la transformada DCT-2.

[0216] Ejemplo 9. El método del ejemplo 8, en donde seleccionar diferentes combinaciones de la transformada DST-7 y la transformada DCT-2 comprende: seleccionar la transformada DST-7 para cualquier fila o columna con menos o igual a K muestras; y seleccionar la transformada DCT-2 para cualquier fila o columna con más de K muestras.

[0217] Ejemplo 10. El método de cualquier combinación de los ejemplos 1-9, que

comprende, además: en respuesta a determinar que el bloque de video está dividido en una pluralidad de particiones, seleccionar los tipos de transformación respectivos para los bloques de coeficientes de cada una de la pluralidad de particiones con el uso de un conjunto de reglas común.

[0218] Ejemplo 11. El método del ejemplo 10, en donde la selección de los tipos de transformación respectivos para cada una de la pluralidad de particiones comprende seleccionar la DST-7 para bloques de coeficientes de toda la pluralidad de particiones.

[0219] Ejemplo 12. El método del ejemplo 10, en donde la selección de tipos de transformación respectivos para cada una de la pluralidad de particiones comprende seleccionar diferentes combinaciones de la transformada DST-7 y la transformada DCT-2 en función de las dimensiones de las particiones.

[0220] Ejemplo 13. El método del ejemplo 12, en donde seleccionar diferentes combinaciones de la transformada DST-7 y la transformada DCT-2 en función de las dimensiones de las particiones comprende: seleccionar la transformada DST-7 para cualquier fila o columna con menos de un número umbral de muestras o igual a un número umbral de muestras; y seleccionar la transformada DCT-2 para cualquier fila o columna con un número de muestras mayor que el umbral.

[0221] Ejemplo 14. El método del ejemplo 13, en donde el umbral es 16.

[0222] Ejemplo 15. El método de cualquier combinación de los ejemplos 10-14, en donde dividir el bloque de video en la pluralidad de particiones comprende dividir el bloque de video mediante particiones intrasub-bloques (ISP).

[0223] Ejemplo 16. El método del ejemplo 15, en donde la transformación con el uso de la transformada DST-7 comprende transformar el bloque de coeficiente actual con el uso de la siguiente matriz DST-7 de 2 puntos:

$$\begin{Bmatrix} 48, & 77 \\ 77, & -48 \end{Bmatrix}.$$

[0224] Ejemplo 17. El método de cualquier combinación de los ejemplos 1-16, que comprende, además: determinar un modo de intrapredicción utilizado para predecir el bloque de video, en donde derivar el tipo de transformación para el bloque de coeficiente actual del bloque de video comprende derivar el tipo de transformación para el bloque de coeficiente actual del bloque de video basado en el modo de intrapredicción.

[0225] Ejemplo 18. El método del ejemplo 17, en donde derivar el tipo de transformación para el bloque de coeficiente actual del bloque de video en base al modo de intrapredicción comprende: en respuesta a la determinación de que el modo de intrapredicción es un modo plano o DC, seleccionar la transformada DST-7 para el

bloque de coeficiente actual en ambas direcciones horizontal y vertical.

[0226] Ejemplo 19. El método de cualquiera de los ejemplos 17 o 18, en donde derivar el tipo de transformación para el bloque de coeficiente actual del bloque de video basado en el modo de intrapredicción comprende: en respuesta a la determinación de que el modo de intrapredicción es un modo diagonal angular, seleccionar la transformada DST-7 para el bloque de coeficiente actual en ambas direcciones horizontal y vertical.

[0227] Ejemplo 20. El método del ejemplo 19, en donde el modo angular diagonal es el índice de modo 34.

[0228] Ejemplo 21. El método de cualquiera de los ejemplos 17-20, en donde derivar el tipo de transformación para el bloque de coeficiente actual del bloque de video en base al modo de intrapredicción comprende: en respuesta a la determinación de que el modo de intrapredicción es un modo angular, seleccionar el tipo de transformación para el bloque de coeficientes actual en función de un índice de modo del modo de intrapredicción.

[0229] Ejemplo 22. El método del ejemplo 21, en donde la selección del tipo de transformación para el bloque de coeficiente actual en función del índice de modo del modo de intrapredicción comprende: identificar un rango de una pluralidad de rangos que incluye índice de modo del modo de intrapredicción; y seleccionar el tipo de transformación para el bloque de coeficiente actual en base al rango identificado.

[0230] Ejemplo 23. El método del ejemplo 22, en donde identificar el rango comprende: identificar un primer rango en respuesta a determinar que el índice de modo está entre un primer umbral y un segundo umbral; identificar un segundo rango en respuesta a la determinación de que el índice de modo está entre el segundo umbral y un tercer umbral; e identificar un tercer rango en respuesta a la determinación de que el índice de modo está entre el tercer umbral y un cuarto umbral.

[0231] Ejemplo 24. El método del ejemplo 23, en donde: identificar el primer rango en respuesta a determinar que el índice de modo está entre el primer umbral y el segundo umbral comprende identificar el primer rango en respuesta a determinar que el índice de modo está dentro de $[2, \dots, (33 - T)]$; identificar el segundo rango en respuesta a determinar que el índice de modo está entre el segundo umbral y el tercer umbral comprende identificar el segundo rango en respuesta a determinar que el índice de modo está dentro de $[(34 - T), \dots, (34 + T)]$; identificar un tercer rango en respuesta a determinar que el índice de modo está entre el tercer umbral y el cuarto umbral comprende identificar el tercer rango en respuesta a determinar que el índice de modo está dentro de $[(35 + T), \dots, 66]$; y T es un número entero entre 2 y 30.

[0232] Ejemplo 25. El método del ejemplo 23 o ejemplo 24, en donde seleccionar el tipo de transformación para el bloque de coeficiente actual en función del rango identificado comprende: seleccionar la DST-7 para uso horizontal y la DCT-2 para uso vertical en respuesta para identificar el primer rango; seleccionar la DST-7 para uso horizontal y vertical en respuesta a la identificación del segundo rango; y seleccionar la DCT-2 para uso horizontal y la DST-7 para uso vertical en respuesta a la identificación del tercer rango.

[0233] Ejemplo 26. El método de cualquiera de los ejemplos 1-25, en donde la codificación comprende la decodificación.

[0234] Ejemplo 27. El método de cualquiera de los ejemplos 1-26, en donde la codificación comprende la codificación.

[0235] Ejemplo 28. Un dispositivo para codificar datos de video, que comprende uno o más medios de ejemplo para realizar el método de cualquiera de los ejemplos 1-27.

[0236] Ejemplo 29. El dispositivo del ejemplo 28, en donde el medio o los medios comprenden uno o más procesadores implementados en circuitos.

[0237] Ejemplo 30. El dispositivo de cualquiera de los ejemplos 28 y 29, que comprende además una memoria para almacenar los datos de video.

[0238] Ejemplo 31. El dispositivo de cualquiera de los ejemplos 28-30, que comprende además una pantalla configurada para mostrar datos de video decodificados.

[0239] Ejemplo 32. El dispositivo de cualquiera de los ejemplos 28-31, en donde el dispositivo comprende uno o más de una cámara, una computadora, un dispositivo móvil, un dispositivo receptor de transmisión o un decodificador.

[0240] Ejemplo 33. El dispositivo de cualquiera de los ejemplos 28-32, en donde el dispositivo comprende un decodificador de video.

[0241] Ejemplo 34. El dispositivo de cualquiera de los ejemplos 28-33, en donde el dispositivo comprende un codificador de video.

[0242] Ejemplo 35. Un medio de almacenamiento legible por computadora que tiene almacenadas instrucciones que, cuando se ejecutan, hacen que uno o más procesadores realicen el método de cualquiera de los ejemplos 1-25.

[0243] Debe reconocerse que, dependiendo del ejemplo, ciertos actos o eventos de cualquiera de las técnicas descritas en el presente documento pueden realizarse en una secuencia diferente, pueden agregarse, fusionarse u omitirse por completo (por ejemplo, no todos los actos o eventos descritos son necesarios para la práctica de las técnicas). Además, en ciertos ejemplos, los actos o eventos pueden realizarse simultáneamente, por ejemplo, a través de procesamiento de subprocesos múltiples, procesamiento de

interrupción o procesadores múltiples, en lugar de secuencialmente.

[0244] En uno o más ejemplos, las funciones descritas pueden implementarse en *hardware*, *software*, *firmware* o cualquier combinación de estos. Si se implementa en *software*, las funciones pueden almacenarse o transmitirse como una o más instrucciones o códigos en un medio legible por computadora y ejecutarse por una unidad de procesamiento basada en *hardware*. Los medios legibles por computadora pueden incluir medios de almacenamiento legibles por computadora, que corresponden a un medio tangible como los medios de almacenamiento de datos o medios de comunicación que incluyen cualquier medio que facilite la transferencia de un programa de computadora de un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por computadora generalmente pueden corresponder a (1) medios tangibles de almacenamiento legibles por computadora que no son transitorios o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que puedan acceder una o más computadoras o uno o más procesadores para recuperar instrucciones, códigos y/o estructuras de datos para la implementación de las técnicas descritas en esta divulgación. Un producto de programa de computadora puede incluir un medio legible por computadora.

[0245] A modo de ejemplo, y sin limitación, dichos medios de almacenamiento legibles por computadora pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento en disco óptico, almacenamiento en disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que puede usarse para almacenar el código de programa deseado en forma de instrucciones o estructuras de datos y al que puede acceder una computadora. Además, cualquier conexión se denomina correctamente un medio legible por computadora. Por ejemplo, si las instrucciones se transmiten desde un sitio web, un servidor u otra fuente remota con el uso de un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o tecnologías inalámbricas como infrarrojos, radio y microondas, entonces el cable coaxial, cable de fibra óptica, par trenzado, DSL o tecnologías inalámbricas como infrarrojos, radio y microondas están incluidos en la definición de medio. Sin embargo, debe entenderse que los medios de almacenamiento legibles por computadora y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que se dirigen a medios de almacenamiento tangibles no transitorios. Disco, como se usa en este documento, incluye disco compacto (CD), disco láser, disco óptico, disco versátil digital (DVD),

disquete y disco Blu-ray, donde los discos generalmente reproducen datos magnéticamente, mientras que los discos reproducen datos ópticamente con láser. Las combinaciones de lo anterior también deben incluirse dentro del alcance de los medios legibles por computadora.

[0246] Las instrucciones pueden ser ejecutadas por uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados específicos de aplicación (ASIC), matrices de puertas programables en campo (FPGA), u otros equivalentes integrados o circuito lógico discreto. Por consiguiente, los términos “procesador” y “circuito de procesamiento”, como se usan en el presente documento, pueden hacer referencia a cualquiera de las estructuras anteriores o cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en este documento puede proporcionarse dentro de módulos dedicados de *hardware* y/o *software* configurados para codificación y decodificación, o incorporados en un códec combinado. Además, las técnicas podrían implementarse completamente en uno o más circuitos o elementos lógicos.

[0247] Las técnicas de la presente divulgación pueden implementarse en una amplia variedad de dispositivos o aparatos, incluyendo un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En esta divulgación se describen diversos componentes, módulos o unidades para enfatizar los aspectos funcionales de los dispositivos configurados para realizar las técnicas divulgadas, pero no necesariamente requieren la realización por diferentes unidades de *hardware*. Por el contrario, como se describió con anterioridad, varias unidades pueden combinarse en una unidad de *hardware* de códec o proporcionarse mediante una colección de unidades de *hardware* interoperativas, que incluyen uno o más procesadores como se describe con anterioridad, junto con el *software* y/o *firmware* adecuado.

[0248] Se han descrito varios ejemplos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.